AudioCodes Mediant™ Media Gateways

SIP

Mediant 1000 & Mediant 600

User's Manual

Version 5.4

Document #: LTRT-83303

May 2008

Table of Contents

1 Overview 19

1.1 Mediant 1000 19
1.2 Mediant 600 20
1.3 SIP Overview 21

2 Configuration Concepts 23

3 Web-Based Management 25

3.1 Computer Requirements 25
3.2 Accessing the Web Interface 26
3.3 Getting Acquainted with the Web Interface 27
3.3.1 Toolbar 28
3.3.2 Navigation Tree 29
3.3.2.1 Displaying Navigation Tree in Basic and Full View 30
3.3.2.2 Showing / Hiding the Navigation Pane 31
3.3.3 Working with Configuration Pages 31
3.3.3.1 Accessing Pages 32
3.3.3.2 Viewing Parameters 32
3.3.3.3 Modifying and Saving Parameters 34
3.3.3.4 Entering Phone Numbers in Various Tables 36
3.3.3.5 Working with Tables 36
3.3.4 Searching for Configuration Parameters 38
3.3.5 Working with Scenarios 40
3.3.5.1 Creating a Scenario 40
3.3.5.2 Accessing a Scenario 42
3.3.5.3 Editing a Scenario 43
3.3.5.4 Saving a Scenario to a PC 44
3.3.5.5 Loading a Scenario to the Device 46
3.3.5.6 Deleting a Scenario 46
3.3.5.7 Exiting Scenario Mode 47
3.3.6 Customizing the Web Interface 48
3.3.6.1 Replacing the Corporate Logo 48
3.3.6.2 Customizing the Product Name 50
3.3.6.3 Creating a Login Welcome Message 51
3.3.7 Getting Help 52
3.3.8 Using the Home Page 52
3.3.8.1 Assigning a Name to a Port 57
3.3.8.2 Viewing Analog Port Information 58
3.3.8.3 Resetting an Analog Channel 59
3.3.8.4 Viewing Trunks' Channels 59
3.3.8.5 Replacing Modules 61
3.3.9 Logging Off the Web Interface 62
3.4 Configuration Tab 64
3.4.1 Network Settings 64
3.4.1.1 Configuring the IP Settings 64
3.4.1.2 Configuring the Multiple Interface Table 68
3.4.1.3 Configuring the Application Settings 71
3.4.1.4 Configuring the NFS Settings 74
3.4.1.5 Configuring the IP Routing Table 75
3.4.1.6 Configuring the QoS Settings 78

3.4.2 Media Settings 80
3.4.2.1 Configuring the Voice Settings 81
3.4.2.2 Configuring the Fax / Modem / CID Settings 84
3.4.2.3 Configuring the RTP / RTCP Settings 88
3.4.2.4 Configuring the IPmedia Settings 92
3.4.2.5 Configuring the General Media Settings 93
3.4.2.6 Configuring the Hook-Flash Settings 94
3.4.3 PSTN Settings 96
3.4.3.1 Configuring the Trunk Settings 96
3.4.3.2 Configuring the CAS State Machines 109
3.4.4 Security Settings 111
3.4.4.1 Configuring the Web User Accounts 112
3.4.4.2 Configuring the Web and Telnet Access List 114
3.4.4.3 Configuring the Firewall Settings 116
3.4.4.4 Configuring the Certificates 118
3.4.4.5 Configuring the General Security Settings 123
3.4.4.6 Configuring the IPSec Table 126
3.4.4.7 Configuring the IKE Table 130
3.4.5 Protocol Configuration 133
3.4.5.1 Configuring the Protocol Definition Parameters 133
3.4.5.2 Configuring the SIP Advanced Parameters 165
3.4.5.3 Configuring the Number Manipulation Tables 190
3.4.5.4 Configuring the Routing Tables 197
3.4.5.5 Configuring the Profile Definitions 212
3.4.5.6 Configuring the Endpoint Settings 218
3.4.5.7 Configuring the Trunk and IP Groups 226
3.4.5.8 Configuring the Digital Gateway Parameters 235
3.4.5.9 Configuring the IPmedia Parameters 242
3.4.6 Advanced Applications 244
3.4.6.1 Configuring the Voice Mail (VM) Parameters 244
3.4.6.2 Configuring RADIUS Accounting Parameters 247
3.4.6.3 Configuring the FXO Parameters 248
3.4.7 Configuring the TDM Bus Settings 251
3.5 Management Tab 253
3.5.1 Management Configuration 253
3.5.1.1 Configuring the Management Settings 254
3.5.1.2 Configuring the Regional Settings 261
3.5.1.3 Maintenance Actions 262
3.5.2 Software Update 266
3.5.2.1 Loading Auxiliary Files 266
3.5.2.2 Updating the Software Upgrade Key 268
3.5.2.3 Software Upgrade Wizard 272
3.5.2.4 Backing Up and Restoring Configuration 276
3.6 Status & Diagnostics Tab 278
3.6.1 Status & Diagnostics 278
3.6.1.1 Viewing the Device's Syslog Messages 278
3.6.1.2 Viewing the Ethernet Port Information 280
3.6.1.3 Viewing Device Information 280
3.6.1.4 Viewing Performance Statistics 282
3.6.1.5 Viewing Active Alarms 282
3.6.2 Gateway Statistics 283
3.6.2.1 Call Counters 283
3.6.2.2 Call Routing Status 286
3.6.2.3 Registration Status 287
3.6.2.4 SAS/SBC Registered Users 288
3.6.2.5 IP Connectivity 289

4 ini File Configuration 291

4.1 Secured Encoded ini File 291
4.2 The ini File Structure 292
4.2.1 Structure Rules 292
4.2.2 Structure of Individual ini File Parameters 292
4.2.3 Structure of ini File Table Parameters 293
4.2.4 Example of an ini File 295
4.3 Modifying an ini File 296
4.4 Reference for ini File Parameters 296
4.4.1 Networking Parameters 296
4.4.2 System Parameters 305
4.4.3 Web and Telnet Parameters 311
4.4.4 Security Parameters 314
4.4.5 RADIUS Parameters 320
4.4.6 SNMP Parameters 321
4.4.7 SIP Configuration Parameters 323
4.4.8 Media Server Parameters 342
4.4.9 Voice Mail Parameters 343
4.4.10 PSTN Parameters 344
4.4.11 ISDN and CAS Interworking-Related Parameters 348
4.4.12 Analog Telephony Parameters 355
4.4.13 Number Manipulation and Routing Parameters 364
4.4.14 Channel Parameters 377
4.4.15 Auxiliary / Configuration Files Parameters 383

5 Default Settings 385

5.1 Defining Default Settings 385
5.2 Restoring Factory Defaults 385

6 Auxiliary Configuration Files 387

6.1 Configuring the Call Progress Tones File 387
6.2 Configuring the Distinctive Ringing Section of the ini File 390
6.2.1 Examples of Ringing Signals 391
6.3 Prerecorded Tones (PRT) File 392
6.4 CAS Protocol Auxiliary Files 392
6.5 Coefficient File 393
6.6 Dial Plan File 393
6.7 User Information File 395

7 IP Telephony Capabilities 397

7.1 Configuring the DTMF Transport Types 397
7.2 Fax and Modem Capabilities 398
7.2.1 Fax/Modem Operating Modes 398
7.2.2 Fax/Modem Transport Modes 399
7.2.2.1 T.38 Fax Relay Mode 399
7.2.2.2 Fax/Modem Bypass Mode 400
7.2.2.3 Fax / Modem NSE Mode 401
7.2.2.4 G.711 Fax / Modem Transport Mode 402
7.2.2.5 Fax Fallback 402
7.2.2.6 Fax / Modem Transparent Mode 403
7.2.2.7 Fax / Modem Transparent with Events Mode 404

7.2.3 Supporting V.34 Faxes 404
7.2.3.1 Using Bypass Mechanism for V.34 Fax Transmission 404
7.2.3.2 Using Relay mode for both T.30 and V.34 faxes 405
7.2.4 Supporting V.152 Implementation 405
7.3 FXO Operating Modes 406
7.3.1 IP-to-Telephone Calls 406
7.3.1.1 One-Stage Dialing 406
7.3.1.2 Two-Stage Dialing 408
7.3.1.3 Call Termination (Disconnect Supervision) on FXO Devices 408
7.3.1.4 DID Wink 409
7.3.2 Telephone-to-IP Calls 410
7.3.2.1 Automatic Dialing 410
7.3.2.2 Collecting Digits Mode 411
7.3.2.3 Ring Detection Timeout 411
7.3.2.4 FXO Supplementary Services 412
7.4 Event Notification using X-Detect Header 412
7.5 RTP Multiplexing (ThroughPacket) 414
7.6 Dynamic Jitter Buffer Operation 414
7.7 Configuring Alternative Routing (Based on Connectivity and QoS) 415
7.7.1 Alternative Routing Mechanism 415
7.7.2 Determining the Availability of Destination IP Addresses 416
7.7.3 PSTN Fallback as a Special Case of Alternative Routing 416
7.7.4 Relevant Parameters 416
7.8 Mapping PSTN Release Cause to SIP Response 416
7.9 Call Detail Record 417
7.10 Supported RADIUS Attributes 419
7.11 Trunk-to-Trunk Routing Example 421
7.12 Proxy or Registrar Registration Example 422
7.13 Configuration Examples 423
7.13.1 SIP Call Flow 423
7.13.2 SIP Authentication Example 426
7.13.3 Establishing a Call between Two Devices 428
7.13.4 Remote PBX Extension Between FXO and FXS Devices 429
7.13.4.1 Dialing from Remote Extension (Phone at FXS) 430
7.13.4.2 Dialing from PBX Line or PSTN 431
7.13.4.3 Message Waiting Indication for Remote Extensions 431
7.13.4.4 Call Waiting for Remote Extensions 432
7.13.4.5 FXS Gateway Configuration 432
7.13.4.6 FXO Gateway Configuration 433
7.13.5 SIP Trunking between Enterprise and ITSPs 434
7.14 Working with Supplementary Services 437
7.14.1 Call Hold and Retrieve 437
7.14.2 Consultation / Alternate 438
7.14.3 Call Transfer 438
7.14.4 Call Forward 439
7.14.5 Call Waiting 440
7.14.6 Message Waiting Indication 440
7.14.7 Caller ID 441
7.14.7.1 Caller ID Detection / Generation on the Tel Side 441
7.14.7.2 Debugging a Caller ID Detection on FXO 442
7.14.7.3 Caller ID on the IP Side 443

8 Networking Capabilities 445

8.1 Ethernet Interface Configuration 445
8.2 Ethernet Interface Redundancy 445
8.3 NAT (Network Address Translation) Support 446
8.3.1 STUN 447
8.3.2 First Incoming Packet Mechanism 448
8.3.3 No-Op Packets 448
8.4 IP Multicasting 448
8.5 Robust Reception of RTP Streams 449
8.6 Multiple Routers Support 449
8.7 Simple Network Time Protocol Support 449
8.8 IP QoS via Differentiated Services (DiffServ) 450
8.9 VLANS and Multiple IPs 450
8.9.1 Multiple IPs 450
8.9.2 IEEE 802.1p/Q (VLANs and Priority) 451
8.9.3 Getting Started with VLANS and Multiple IPs 453
8.9.3.1 Integrating Using the Web Interface 454
8.9.3.2 Integrating Using the ini File 456

9 Advanced PSTN Configuration 459

9.1 Clock Settings 459
9.2 Release Reason Mapping 459
9.2.1 Reason Header 460
9.2.2 Fixed Mapping of ISDN Release Reason to SIP Response 460
9.2.3 Fixed Mapping of SIP Response to ISDN Release Reason 462
9.3 ISDN Overlap Dialing 463
9.4 Using ISDN NFAS 464
9.4.1 NFAS Interface ID 465
9.4.2 Working with DMS-100 Switches 466
9.4.3 Creating an NFAS-Related Trunk Configuration On-The-Fly 466
9.5 Redirect Number and Calling Name (Display) 468

10 Media Server Capabilities 469

10.1 Conference Server 469
10.1.1 Simple Conferencing (NetAnn) 470
10.1.1.1 SIP Call Flow 470
10.1.1.2 Creating a Conference 470
10.1.1.3 Joining a Conference 471
10.1.1.4 Terminating a Conference 471
10.1.1.5 PSTN Participants 472
10.1.2 Advanced Conferencing (MSCML) 472
10.1.2.1 Creating a Conference 472
10.1.2.2 Joining a Conference 473
10.1.2.3 Modifying a Conference 474
10.1.2.4 Applying Media Services on a Conference 474
10.1.2.5 Active Speaker Notification 475
10.1.2.6 Terminating a Conference 476
10.1.3 Conference Call Flow Example 477

10.2 Announcement Server 483
10.2.1 NetAnn Interface 483
10.2.1.1 Playing a Local Voice Prompt 483
10.2.1.2 Playing using HTTP/NFS Streaming 483
10.2.1.3 Supported Attributes 484
10.2.2 MSCML Interface 484
10.2.2.1 Operation 485
10.2.2.2 Playing Announcements 488
10.2.2.3 Playing Announcements and Collecting Digits 488
10.2.2.4 Playing Announcements and Recording Voice 490
10.2.2.5 Stopping the Playing of an Announcement 491
10.2.2.6 Relevant Parameters 491
10.2.3 Announcement Call Flow Example 492
10.3 IP-to-IP Transcoding 494

11 Tunneling Applications 497

11.1 TDM Tunneling 497
11.2 QSIG Tunneling 500

12 OSN Server Hardware Installation 501

12.1 OSN Server Modules 501
12.2 Required Working Tools 502
12.3 Installing the CM Module 502
12.4 Installing the iPMX Module 504
12.5 Installing the HDMX Module 506
12.6 Replacing the iPMX Module's Lithium Battery 507

13 Installing Windows XP on the OSN Server 509

13.1 Requirements 509
13.2 Windows XP Installation Procedures 509
13.2.1 Stage 1: Creating the Helper Script 509
13.2.2 Stage 2: Creating the ISO Image 510
13.2.3 Stage 3: Burning the CD 514
13.2.4 Stage 4: Cabling the Mediant 1000 OSN Server 515
13.2.5 Stage 5: Booting and Installing 515
13.2.6 Stage 6: Enabling the Firewall with Remote Desktop Access 517
13.3 Connecting Remotely to OSN Server using Windows XP 519
13.3.1 Cabling 519
13.3.2 Connecting Using Remote Desktop Connection 520

14 Installing Linux™ Operating System on the OSN Server 523

14.1 Requirements 523
14.1.1 Hardware 523
14.1.2 Software 524
14.2 Cabling 524
14.3 Installing Linux™ RedHat (and Fedora) 525
14.3.1 Stage 1: Obtaining the Linux Redhat ISO Image 525
14.3.1.1 Downloading an Updated ISO Image 525
14.3.1.2 Creating an Updated ISO Image 526
14.3.2 Stage 2: Editing the isolinux.cfg File 528
14.3.3 Stage 3: Burning ISO Image File to CD-ROM 532
14.3.4 Stage 4: Installing the Boot Media 532

14.3.5 Additional RedHat™ and Fedora™ Installation Notes 534
14.3.6 Post-installation Notes for Kernels 2.6+ (Fedora™ Core 4+ and RedHat™ EL 4+)534
14.4 Installing Linux™ Debian 535
14.4.1 Stage 1: Obtaining the ISO Image 535
14.4.2 Stage 2: Preparing the Boot Media 536
14.4.3 Stage 3: Editing the isolinux.cfg File 538
14.4.3.1 Downloading an Updated Debian isolinux.cfg File 538
14.4.3.2 Editing the isolinux.cfg File 538
14.4.4 Stage 4: Burning ISO Image to CD 541
14.4.5 Stage 5: Installing the Boot Media 541
14.4.6 Additional Linux™ Debian Installation Notes 542
14.5 Installing Linux™ SUSE 544
14.5.1 Additional Requirement for Linux™ SUSE Installation 544
14.5.2 Stage 1: Obtaining the ISO Image 544
14.5.3 Stage 2: Preparing the Boot Media 545
14.5.4 Stage 3: Editing the isolinux.cfg File 547
14.5.4.1 Downloading an Updated SUSE isolinux.cfg File 547
14.5.4.2 Editing the isolinux.cfg File 548
14.5.5 Stage 4: Burning the CD 551
14.5.6 Stage 5: Installing the Boot Media 552
15 Supplied SIP Software Package 555
16 Selected Technical Specifications 557
16.1 Mediant 1000 557
16.2 Mediant 600 561
17 Glossary 565

List of Figures

Figure 3-1: Enter Network Password Screen 26
Figure 3-2: Main Areas of the Web Interface GUI 27
Figure 3-3: "Reset" Displayed on Toolbar 28
Figure 3-4: Terminology for Navigation Tree Levels 29
Figure 3-5: Navigation Tree in Basic and Full View 30
Figure 3-6: Showing and Hiding Navigation Pane 31
Figure 3-7: Toggling between Basic and Advanced Page View 33
Figure 3-8: Expanding and Collapsing Parameter Groups 34
Figure 3-9: Editing Symbol after Modifying Parameter Value 35
Figure 3-10: Value Reverts to Previous Valid Value 35
Figure 3-11: Adding an Index Entry to a Table 36
Figure 3-12: Compacting a Web Interface Table 37
Figure 3-13: Searched Result Screen 39
Figure 3-14: Scenario Creation Confirm Message Box 40
Figure 3-15: Creating a Scenario 41
Figure 3-16: Scenario Loading Message Box 42
Figure 3-17: Scenario Example 42
Figure 3-18: Scenario File Page 45
Figure 3-19: Scenario Loading Message Box 46
Figure 3-20: Message Box for Confirming Scenario Deletion 47
Figure 3-21: Confirmation Message Box for Exiting Scenario Mode 47
Figure 3-22: Customizing Web Logo and Product Name 48
Figure 3-23: Image Download Screen 49
Figure 3-24: User-Defined Web Welcome Message after Login 51
Figure 3-25: Help Topic for Current Page 52
Figure 3-26: Mediant 1000 Home Page 53
Figure 3-27: Mediant 600 Home Page 53
Figure 3-28: Shortcut Menu (Example, Mediant 1000 Home Page) 57
Figure 3-29: Text Box for Typing Port Name (Example, Mediant 1000 Home Page) 57
Figure 3-30: Selecting Port Settings from Shortcut Menu 58
Figure 3-31: Basic Information Screen 58
Figure 3-32: Resetting Channel 59
Figure 3-33: Trunks & Channels Status Page (Example, Mediant 1000) 59
Figure 3-34: Basic Channel Information Page (Example, Mediant 1000) 60
Figure 3-35: Remove Module Button Appears after Clicking Module Name 61
Figure 3-36: Module Removal Confirmation Message Box 61
Figure 3-37: Removed Module 62
Figure 3-38: Insert Module Button after Clicking Module's Name 62
Figure 3-39: Log Off Confirmation Box 62
Figure 3-40: Web Session Logged Off 63
Figure 3-41: IP Settings Page 65
Figure 3-42: Confirmation Message for Accessing the Multiple Interface Table 68
Figure 3-43: Interface Table Page 68
Figure 3-44: Application Settings Page 71
Figure 3-45: NFS Settings Page 74
Figure 3-46: IP Routing Table Page 76
Figure 3-47: QoS Settings Page 78
Figure 3-48: Voice Settings Page 81
Figure 3-49: Fax/Modem/CID Settings Page 84
Figure 3-50: RTP/RTCP Settings Page 88
Figure 3-51: IPMedia Settings Page 92
Figure 3-52: General Media Settings Page 93
Figure 3-53: Hook-Flash Settings Screen 94
Figure 3-54: Trunk Settings Page 96
Figure 3-55: Trunk Scroll Bar 97
Figure 3-56: CAS State Machine Page 110
Figure 3-57: Web User Accounts Page (for Users with 'Security Administrator' Privileges) 113

Figure 3-58: Web & Telnet Access List Page - Add New Entry 114
Figure 3-59: Web & Telnet Access List Table 115
Figure 3-60: Firewall Settings Page 116
Figure 3-61: Certificates Signing Request Page 119
Figure 3-62: General Security Settings Page 123
Figure 3-63: IPSec Table Page 127
Figure 3-64: IKE Table Page 130
Figure 3-65: SIP General Parameters 134
Figure 3-66: Proxy Sets Table Page 154
Figure 3-67: Coders Page 159
Figure 3-68: DTMF & Dialing Page 160
Figure 3-69: Advanced Parameters Page 166
Figure 3-70: Supplementary Services Page 176
Figure 3-71: Metering Tones Page 182
Figure 3-72: Charge Codes Table Page 183
Figure 3-73: Keypad Features Page 185
Figure 3-74: SAS Configuration Page 189
Figure 3-75: Source Phone Number Manipulation Table for Tel-to-IP Calls 191
Figure 3-76: Phone Context Table Page 196
Figure 3-77: Tel to IP Routing Page 202
Figure 3-78: IP to Trunk Group Routing Table Page 205
Figure 3-79: Internal DNS Table Page 207
Figure 3-80: Internal SRV Table Screen 208
Figure 3-81: Reasons for Alternative Routing Page 210
Figure 3-82: Release Cause Mapping Page 211
Figure 3-83: Coder Group Settings Page 213
Figure 3-84: Tel Profile Settings Page 215
Figure 3-85: IP Profile Settings Page 217
Figure 3-86: Authentication Screen 219
Figure 3-87: Automatic Dialing Page 220
Figure 3-88: Caller Display Information Page 221
Figure 3-89: Call Forward Table Page 223
Figure 3-90: Caller ID Permissions Page 224
Figure 3-91: Caller Waiting Page 225
Figure 3-92: Trunk Group Table Page 226
Figure 3-93: Trunk Group Settings Page 228
Figure 3-94: IP Group Table Page 231
Figure 3-95: Account Table Page 233
Figure 3-96: Digital Gateway Parameters Page 235
Figure 3-97: IPmedia Parameters Page 242
Figure 3-98: Voice Mail Settings Page 244
Figure 3-99: RADIUS Parameters Page 247
Figure 3-100: FXO Settings Page 248
Figure 3-101: TDM Bus Settings Page 251
Figure 3-102: Management Settings Page 254
Figure 3-103: SNMP Trap Destinations Page 257
Figure 3-104: SNMP Community Strings Page 258
Figure 3-105: SNMP V3 Setting Page 259
Figure 3-106: SNMP Trusted Managers 261
Figure 3-107: Regional Settings Page 261
Figure 3-108: Maintenance Actions Page 262
Figure 3-109: Reset Confirmation Message Box 263
Figure 3-110: Device Lock Confirmation Message Box 264
Figure 3-111: Load Auxiliary Files Page 267
Figure 3-112: Software Upgrade Key with Multiple S/N Lines 270
Figure 3-113: Start Software Upgrade Wizard Screen 273
Figure 3-114: Load a CMP File Wizard Page 273
Figure 3-115: Loaded CMP File 274
Figure 3-116: Load an ini File Page 275

Figure 3-117: End Process Wizard Page 276
Figure 3-118: Message Box Informing of Upgraded CMP File 276
Figure 3-119: Configuration File Page 277
Figure 3-120: Message Log Screen 279
Figure 3-121: Ethernet Port Information Page 280
Figure 3-122: Device Information Page 281
Figure 3-123: Performance Statistics Page 282
Figure 3-124: Active Alarms Page 283
Figure 3-125: Calls Count Page 284
Figure 3-126: Call Routing Status Page 286
Figure 3-127: Registration Status Page 287
Figure 3-128: SAS Registered Users Page 288
Figure 3-129: IP Connectivity Page 289
Figure 6-1: Example of a User Information File 395
Figure 7-1: Call Flow for One-Stage Dialing 407
Figure 7-2: Call Flow for Two-Stage Dialing 408
Figure 7-3: Call Flow for Collecting Digits Mode 411
Figure 7-4: SIP Call Flow 423
Figure 7-5: Assigning Phone Numbers to Device 10.2.37.10) 428
Figure 7-6: Assigning Phone Numbers to Device 10.2.37.20 428
Figure 7-7: Routing Calls Between Devices 429
Figure 7-8: FXO-FXS Remote PBX Extension (Example) 430
Figure 7-9: MWI for Remote Extensions 431
Figure 7-10: Call Waiting for Remote Extensions 432
Figure 7-11: Assigning Phone Numbers to FXS Endpoints 432
Figure 7-12: Automatic Dialing for FXS Ports 432
Figure 7-13: Assigning Phone Numbers to FXO Ports 433
Figure 7-14: FXO Automatic Dialing Configuration 433
Figure 7-15: FXO Tel-to-IP Routing Configuration 433
Figure 7-16: Example Setup for Routing Between ITSP and Enterprise PBX 434
Figure 7-17: Configuring Proxy Set ID #1 in the Proxy Sets Table Page 435
Figure 7-18: Configuring IP Groups #1 and #2 in the IP Group Table Page 435
Figure 7-19: Assign the Trunk to Trunk Group ID #1 in the Trunk Group Table Page 436
Figure 7-20: Configuring Trunk Group #1 for Registration in Trunk Group Settings Page 436
Figure 7-21: Configuring Accounts for PBX Registration to ITSPs in Account Table Page 436
Figure 7-22: Configuring ITSP-to-Trunk Group #1 Routing in IP to Trunk Group Table Page 436
Figure 7-23: Configuring Tel-to-IP Routing to ITSPs in Tel to IP Routing Table Page 436
Figure 8-1: Multiple Network Interfaces and VLANs 452
Figure 8-2: VLAN Configuration in the IP Settings Page 454
Figure 8-3: OAM, Control, Media IP Configuration in the IP Settings Page 455
Figure 8-4: Multiple Interface Table Page 455
Figure 8-5: Static Routes for OAM/Control in IP Routing Table 456
Figure 10-1: Simple Conferencing SIP Call Flow 470
Figure 10-2: Advanced Conferencing SIP Call Flow 473
Figure 10-3: Modifying a Conference - SIP Call Flow 474
Figure 10-4: Applying Media Services on a Conference -- SIP Call Flow 475
Figure 10-5: Terminating a Conference -- SIP Call Flow 476
Figure 10-6: Conference Call Flow Example 477
Figure 10-7: MSCML Architecture 485
Figure 10-8: Direct Connection (Example) 495
Figure 10-9: Using an Application Server 496
Figure 12-1: Connection Module (CM) 501
Figure 12-2: iPMX Module 501
Figure 12-3: Hard Drive Module (HDMX) 502
Figure 12-4: Mediant 1000 Front Panel 503
Figure 12-5: Inserting CM Module 503
Figure 12-6: Mediant 1000 Rear Panel 504
Figure 12-7: Mediant 1000 with Cover Plates Removed 504
Figure 12-8: Mediant 1000 with Cutter Tool 505

Figure 12-9: Inserting iPMX Module 505
Figure 12-10: Inserting HDMX Module 506
Figure 12-11: Removing Lithium Battery from iPMX Module 508
Figure 13-1: Selecting Source of Windows XP Installation CD 510
Figure 13-2: Selecting Target Directory for Windows XP 511
Figure 13-3: Defining User and Registration Details 511
Figure 13-4: Creating a User 512
Figure 13-5: Adding LAN and Audio Drivers 512
Figure 13-6: Adding cmd Files Directory to CD 513
Figure 13-7: Selecting do.cmd File 513
Figure 13-8: Adding cmd File to Installation Script 514
Figure 13-9: Completing ISO Image Creation 514
Figure 13-10: Mediant 1000 Front Panel OSN Server Connections 515
Figure 13-11: Windows Security Center 517
Figure 13-12: Remote Desktop Firewall 518
Figure 13-13: Allowing Internet Users Access to Remote Desktop 518
Figure 13-14: Cabling iPMX for Remote Connection from PC with Windows XP 519
Figure 13-15: Changing the PC's IP Address 520
Figure 13-16: Entering IP Address in Remote Desktop Connection 521
Figure 13-17: Entering User Name and Password in Remote Desktop Connection 521
Figure 14-1: Mediant 1000 Front Panel OSN Server Connections 524
Figure 14-2: Disk 1 of Redhat Partner Installation 526
Figure 14-3: Images Folder 527
Figure 14-4: ISO Screen 527
Figure 14-5: Selecting Extract Option 528
Figure 14-6: Extracting Files to Partner Install Folder 528
Figure 14-7: ISO-Extract Screen 529
Figure 14-8: Text Edit Screen 529
Figure 14-9: Deleting CFG 531
Figure 14-10: File Add 531
Figure 14-11: ISO Open Function 532
Figure 14-12: Choose a Language 533
Figure 14-13: WinISO - Actions Screen 535
Figure 14-14: Create ISO from CD-ROM 536
Figure 14-15: Creating .iso File 536
Figure 14-16: Partner Install Folder 537
Figure 14-17: Extract isolinux.cfg 537
Figure 14-18: Extracting Files to Partner Install Folder 537
Figure 14-19: Deleting CFG 540
Figure 14-20: File Add 540
Figure 14-21: ISO Open Function 541
Figure 14-22: WinISO - Actions Screen 544
Figure 14-23: Create ISO from CD-ROM 545
Figure 14-24: Creating .iso File 545
Figure 14-25: Partner Install Folder 546
Figure 14-26: Extract isolinux.cfg File 546
Figure 14-27: Extracting Files to Partner Install Folder 546
Figure 14-28: isolinux.cfg File 548
Figure 14-29: Deleting CFG File 550
Figure 14-30: Add CFG File 550
Figure 14-31: Partner Install Folder 551
Figure 14-32: Save boot.iso 551

List of Tables

Table 3-1: Description of Toolbar Buttons....28
Table 3-2: ini File Parameters for Changing Logo Image....50
Table 3-3: ini File Parameters for Replacing Logo with Text....50
Table 3-4: ini File Parameters for Customizing Product Name....51
Table 3-5: ini File Parameter for Welcome Login Message....51
Table 3-6: Description of the Areas of the Mediant 1000 Home Page....54
Table 3-7: Description of the Areas of the Mediant 600 Home Page....56
Table 3-8: Color-Coding Status for Trunk's Channels....60
Table 3-9: Network Settings -- IP Settings Parameters....66
Table 3-10: Multiple Interface Table Parameters Description....69
Table 3-11: Application Settings Parameters....72
Table 3-12: Network Settings -- NFS Settings Parameters....75
Table 3-13: IP Routing Table Description....76
Table 3-14: QoS Settings Parameters....78
Table 3-15: Media Settings, Voice Settings Parameters....81
Table 3-16: Media Settings -- Fax/Modem/CID Parameters....84
Table 3-17: Media Settings, RTP / RTCP Parameters....89
Table 3-18: IPMedia Parameters....92
Table 3-19: Media Settings Parameters....94
Table 3-20: Hook-Flash Settings Parameters....95
Table 3-21: Trunk (E1/T1/J1/BRI) Configuration Parameters....98
Table 3-22: CAS State Machine Parameters Description....111
Table 3-23: Web User Accounts Access Levels and Privileges....112
Table 3-24: Default Attributes for the Web User Accounts....112
Table 3-25: Internal Firewall Parameters....117
Table 3-26: General Security Parameters....124
Table 3-27: Default IKE Second Phase Proposals....127
Table 3-28: IPSec SPD Table Configuration Parameters....128
Table 3-29: Default IKE First Phase Proposals....131
Table 3-30: IKE Table Configuration Parameters....131
Table 3-31: SIP General Parameters (Protocol Definition)....135
Table 3-32: Proxy & Registration Parameters....147
Table 3-33: Proxy Sets Table Parameters....154
Table 3-34: Supported Coders....158
Table 3-35: DTMF and Dialing Parameters....161
Table 3-36: Advanced Parameters Description....167
Table 3-37: Supplementary Services Parameters....177
Table 3-38: Metering Tones Parameters....182
Table 3-39: Keypad Features Parameters Description....185
Table 3-40: Stand-Alone Survivability Parameters Description....189
Table 3-41: Number Manipulation Parameters Description....192
Table 3-42: Dialing Plan Notations....194
Table 3-43: NPI/TON Values for ISDN ETSI....195
Table 3-44: Phone-Context Parameters Description....196
Table 3-45: Routing General Parameters Description....198
Table 3-46: Tel to IP Routing Table Parameters Description....202
Table 3-47: IP to Trunk Group Routing Table Description....205
Table 3-48: Description of Parameters Unique to IP Profile....218
Table 3-49: Call Forward Table....223
Table 3-50: Trunk Group Table Description....227
Table 3-51: Trunk Group Settings Parameters Description....228
Table 3-52: IP Group Parameters Description....232
Table 3-53: Account Parameters Description....233
Table 3-54: Digital Gateway Parameters Description....236
Table 3-55: IP Media Configuration Parameters....242
Table 3-56: Voice Mail Parameters....245
Table 3-57: RADIUS Parameters Description....247

Table 3-58: FXO Parameters Description 249
Table 3-59: TDM Bus Settings Parameters Description 252
Table 3-60: Management Settings Parameters 255
Table 3-61: SNMP Trap Destinations Parameters Description 257
Table 3-62: SNMP Community Strings Parameters Description 259
Table 3-63: SNMP V3 Users Parameters 260
Table 3-64: Auxiliary Files Descriptions 266
Table 3-65: Ethernet Port Information Parameters 280
Table 3-66: Call Counters Description 284
Table 3-67: Call Routing Status Parameters 286
Table 3-68: SAS Registered Users Parameters 288
Table 3-69: IP Connectivity Parameters 289
Table 4-1: Networking Parameters 296
Table 4-2: System Parameters 305
Table 4-3: Web and Telnet Parameters 311
Table 4-4: Security Parameters 314
Table 4-5: RADIUS Parameters 320
Table 4-6: SNMP Parameters 321
Table 4-7: SIP Parameters 323
Table 4-8: Media Server Parameters 342
Table 4-9: Voice Mail Parameters 343
Table 4-10: PSTN Parameters 344
Table 4-11: ISDN and CAS Interworking-Related Parameters 348
Table 4-12: Analog Telephony Parameters 355
Table 4-13: Number Manipulation and Routing Parameters 364
Table 4-14: Channel Parameters 377
Table 4-15: Auxiliary / Configuration Files Parameters 383
Table 6-1: User Information Items 395
Table 7-1: Supported X-Detect Event Types 412
Table 7-2: Supported CDR Fields 417
Table 7-3: Supported RADIUS Attributes 419
Table 8-1: Traffic / Network Types and Priority 452
Table 8-2: Example of VLAN and Multiple IPs Configuration 454
Table 9-1: Mapping of ISDN Release Reason to SIP Response 460
Table 9-2: Mapping of SIP Response to ISDN Release Reason 462
Table 9-3: Calling Name (Display) 468
Table 9-4: Redirect Number 468
Table 15-1: Supplied Software Package 555
Table 16-1: Mediant 1000 Functional Specifications 557
Table 16-2: Mediant 600 Functional Specifications 561
Table 17-1: Glossary of Terms 565

Reader's Notes

Notice

This document describes the AudioCodes Mediant 1000 and Mediant 600 Voice-over-IP (VoIP) SIP media gateways.

Information contained in this document is believed to be accurate and reliable at the time of printing. However, due to ongoing product improvements and revisions, AudioCodes cannot guarantee accuracy of printed material after the Date Published nor can it accept responsibility for errors or omissions. Before consulting this Manual, check the corresponding Release Notes regarding feature preconditions and/or specific support in this release. In cases where there are discrepancies between this Manual and the Release Notes, the information in the Release Notes supersedes that in this Manual. Updates to this document and other documents can be viewed by registered Technical Support customers at http://www.audiocodes.com, under Support / Product Documentation.

© Copyright 2008 AudioCodes Ltd. All rights reserved.

This document is subject to change without notice.

Date Published: May-09-2008 Date Printed: May-11-2008

Tip: When viewing this manual on CD, Web site or on any other electronic copy, all cross-references are hyperlinked. Click on the page or section numbers (shown in blue) to reach the individual cross-referenced item directly. To return back to the point from where you accessed the cross-reference, press the **ALT** and ← keys

Trademarks

AC logo, Ardito, AudioCoded, AudioCodes, AudioCodes logo, CTI², CTI Squared, InTouch, IPmedia, Mediant, MediaPack, MP-MLQ, NetCoder, Netrake, Nuera, Open Solutions Network, OSN, Stretto, 3GX, TrunkPack, VoicePacketizer, VoIPerfect, What's Inside Matters, Your Gateway To VoIP, are trademarks or registered trademarks of AudioCodes Limited. All other products or trademarks are property of their respective owners.

WEEE EU Directive

Pursuant to the WEEE EU Directive, electronic and electrical waste must not be disposed of with unsorted waste. Please contact your local recycling authority for disposal of this product.

Customer Support

Customer technical support and service are provided by AudioCodes' Distributors, Partners, and Resellers from whom the product was purchased. For Customer support for products purchased directly from AudioCodes, contact support@audiocodes.com.

Abbreviations and Terminology

Each abbreviation, unless widely used, is spelled out in full when first used. Only industry-standard terms are used throughout this manual. Hexadecimal notation is indicated by 0x preceding the number.

Related Documentation

Document #	Manual Name
LTRT-523xx (where *xx* is the document version)	Product Reference Manual
LTRT-831xx	Mediant 1000 and Mediant 600 SIP Release Notes
LTRT-835xx	Mediant 1000 MEGACO-SIP Installation Manual
LTRT-856xx	Mediant 600 SIP Installation Manual
LTRT-665xx	CPE Configuration Guide for IP Voice Mail

Notes: The following naming conventions are used throughout this manual, unless otherwise specified:

- The term *device* refers to the Mediant 1000 and Mediant 600 gateways.
- The term *Trunk* is used synonymously with *Hunt*. Trunk typically refers to digital modules, while Hunt typically refers to analog modules.

Note: Where 'network' appears in this manual, it means Local Area Network (LAN), Wide Area Network (WAN), etc. accessed via the device's Ethernet interface.

1 Overview

This section provides an overview of the Mediant 1000 and Mediant 600 media gateways.

1.1 Mediant 1000

The AudioCodes Mediant 1000 (hereafter referred to as *device*) is a best-of-breed Voice-over-IP (VoIP) Session Initiation Protocol (SIP) Media Gateway, using field-proven, market-leading technology, implementing analog and digital cutting-edge technology. The device is designed to seamlessly interface between Time-Division Multiplexing (TDM) and Internet Protocol (IP) networks, providing superior voice quality and optimized packet voice streaming (voice, fax, and data traffic) over IP networks.

The device is best suited for small-to-medium sized (SME) enterprises, branch offices, and residential media gateway solutions. The device is a highly scalable and modular system that matches the density requirements for smaller environments, while meeting service providers' demands for growth.

The device is ideal for connecting an enterprise's legacy telephones, fax machines, and Private Branch Exchange (PBX) systems to IP-based telephony networks, as well as for seamlessly connecting IP-based PBX architecture to the Public Switched Telephone Network (PSTN). The device also provides SIP trunking capabilities for Enterprises operating with multiple Internet Telephony Service Providers (ITSP) for VoIP services. In addition to operating as a pure media gateway, the device incorporates an open platform, known as the Open Solutions Network (OSN) server, allowing additional deployment options by hosting third-party partner VoIP applications such as IP-PBX, Calling Card, and IP-PBX redundancy.

The device also provides conferencing services over VoIP networks. This is supported by an optional Media Processing Module (MPM) that can be housed in the device's chassis.

The device is fully interoperable with multiple vendor gateways, softswitches, SIP servers, gatekeepers, proxy servers, IP phones, session border controllers (SBC), and firewalls. The device is designed to meet regulatory approval (including Safety, EMC, and Telecom for USA, EU and other countries).

Intelligently packaged in a stackable and compact 1U chassis, it can be mounted on a desk, a wall, or in a standard 19-inch rack. The device is supplied with two integral mounting brackets for facilitating rack installation.

The device is equipped with two 10/100Base-TX Ethernet ports for connection to the IP network. The second Ethernet port is used for 1+1 Ethernet redundancy.

The device supports mixed digital and analog interface configurations:

- **Digital:**
  - The device supports multiples of 1, 2, or 4 E1/T1/J1 spans for connecting the PSTN/PBX to the IP network. The digital modules provide RJ-48 ports. The digital module can be configured with up to 1 or 2 paired spans acting as 'Lifeline' telephone interfaces for switching to the PSTN in case of power or network failure.
  - The device also supports ISDN Basic Rate Interface (BRI) modules for connecting BRI-based PSTN or PBX lines to the IP network. Each BRI module supports four BRI ports (RJ-45). Up to five BRI modules can be housed in the device, supporting up to 20 BRI digital ports. The BRI module can be configured as 'Lifeline' telephone interfaces, switching to the PSTN in case of power failure or network problems.

- **Analog:** The device's analog interface supports up to 24 analog ports (four ports per analog module) in various Foreign Exchange Office (FXO) or Foreign Exchange Station (FXS) configurations, supporting up to 24 simultaneous VoIP calls. The device supports up to six analog modules, each module providing four analog RJ-11 ports. The FXO module can be used to connect analog lines of an enterprise's PBX or the PSTN to the IP network. The FXS module can be used to connect legacy telephones, fax machines, and modems to the IP network. Optionally, the FXS module can be connected to the external trunk lines of a PBX. When deployed with a combination of FXO and FXS modules, the device can be used as a PBX for Small Office Home Office (SOHO) users, and businesses not equipped with a PBX.

The device has enhanced hardware and software capabilities to ease its installation and to maintain voice quality. If the measured voice quality falls beneath a pre-configured value, or the path to the destination is disconnected, the device assures voice connectivity by 'falling' back to the PSTN. In the event of network problems or power failures, calls can be routed back to the PSTN without requiring routing modifications in the PBX. Further reliability is provided by dual Ethernet ports and an optional dual AC power supply.

The device supports various ISDN PRI protocols such as EuroISDN, North American NI2, Lucent™ 4/5ESS, Nortel™ DMS100 and others. It also supports various ISDN BRI protocols such as ETSI 5ESS and QSIG over BRI. In addition, it supports different variants of CAS protocols for E1 and T1 spans, including MFC R2, E&M immediate start, E&M delay dial / start, loop start and ground start.

The device provides a user-friendly, feature-rich embedded HTTP-based Web server for remote configuration and management using a standard Web browser (such as Microsoft™ Internet Explorer™), from anywhere in the world with IP connectivity to the device.

1.2 Mediant 600

AudioCodes' Mediant 600 (hereafter referred to as *device*) is a cost-effective, wireline Voice-over-IP (VoIP) Session Initiation Protocol (SIP)-based media gateway. It is designed to interface between Time-Division Multiplexing (TDM) and IP networks in enterprises, small and medium businesses (SMB), and CPE application service providers. Incorporating AudioCodes' innovative VoIP technology, the device enables rapid time-to-market and reliable cost-effective deployment of next-generation networks.

The device is based on VoIPerfect, AudioCodes underlying, best-of-breed, media gateway core technology. The device provides superior voice technology for connecting legacy telephone and PBX systems to IP networks, as well as seamlessly connecting IP-PBXs to the PSTN. The device is fully interoperable with multiple vendors of IP-PBXs, IP Centrex application servers, softswitches, gateways, proxy servers, IP phones, Session Border Controllers and firewalls.

The device supports one or two E1/T1/J1 spans (including fractional E1/T1), or four to eight ISDN Basic Rate Interface (BRI) ports. The device supports various ISDN PRI protocols such as EuroISDN, North American NI2, Lucent™ 4/5ESS, Nortel™ DMS100 and others, supporting different variants of CAS protocols, including MFC R2, E&M immediate start, E&M delay dial / start, loop- and ground-start signaling. The device also supports various ISDN BRI protocols such as ETSI 5ESS and QSIG over BRI. The device also provides dual Ethernet 10/100Base-TX ports for IP redundancy.

Intelligently packaged in a stackable 1U chassis, the compact device can be mounted on a desk or in a standard 19-inch rack.

The device provides a user-friendly, feature-rich Embedded HTTP-based Web server for remote configuration and management using a standard Web browser (such as Microsoft™ Internet Explorer™), from anywhere in the world with IP connectivity to the device.

1.3 SIP Overview

Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol used on the gateway for creating, modifying, and terminating sessions with one or more participants. These sessions can include Internet telephone calls, media announcements, and conferences.

SIP invitations are used to create sessions and carry session descriptions that enable participants to agree on a set of compatible media types. SIP uses elements called Proxy servers to help route requests to the user's current location, authenticate and authorize users for services, implement provider call-routing policies and provide features to users.

SIP also provides a registration function that enables users to upload their current locations for use by Proxy servers. SIP implemented in the gateway, complies with the Internet Engineering Task Force (IETF) RFC 3261 (refer to http://www.ietf.org).

Reader's Notes

2 Configuration Concepts

You can configure the device's parameters (including upgrading the software, and uploading configuration and auxiliary files), using the following tools:

- An HTTP-based Embedded Web Server (Web interface), using any standard Web browser (described in "Web-based Management" on page 25).
- A configuration file referred to as the *ini* file (refer to "ini File Configuration" on page 291).
- Command Line Interface - CLI (refer to the *Product Reference Manual*).
- Simple Network Management Protocol (SNMP) browser software (refer to the *Product Reference Manual*).
- AudioCodes' Element Management System (refer to *AudioCodes' EMS User's Manual* or *EMS Product Description*).

Note: To initialize the device by assigning it an IP address, a firmware file (*cmp*), and a configuration file (*ini* file), you can use AudioCodes' BootP/TFTP utility, which accesses the device using its MAC address (refer to the *Product Reference Manual*).

Reader's Notes

3 Web-Based Management

The device's Embedded Web Server (*Web interface*) provides FCAPS (fault management, configuration, accounting, performance, and security) functionality. The Web interface allows you to remotely configure your device for quick-and-easy deployment, including uploading of configuration (software upgrade) and auxiliary files, and resetting the device. The Web interface provides real-time, online monitoring of the device, including display of alarms and their severity. In addition, it displays performance statistics of voice calls and related traffic parameters.

The Web interface provides a user-friendly, graphical user interface (GUI), which can be accessed using any standard Web browser (e.g., Microsoft™ Internet Explorer). Access to the Web interface is controlled by various security mechanisms such as login user name and password, read / write privileges, and limiting access to specific IP addresses.

Notes:

- The Web interface allows you to configure most of the device's parameters. Those parameters that are not available in the Web interface can be configured using the *ini* file.
- Certain Web interface pages are feature-key dependant, and therefore, only appear if your device's feature key supports the features relating to these pages (refer to "Updating the Software Upgrade Key" on page 268).
- Throughout this section, parameters enclosed in square brackets [...] depict the *ini* file parameters for configuring the device using the *ini* file.

3.1 Computer Requirements

To use the device's Web interface, the following is required:

- A connection to the Internet network (World Wide Web).
- A network connection to the device's Web interface.
- One of the following Web browsers:
  - Microsoft™ Internet Explorer™ (version 6.0 or later).
  - Netscape™ Navigator™ (version 7.2 or later).
  - Mozilla Firefox® (version 1.5.0.10 or later).
- Recommended screen resolution of 1024 x 768 pixels, or 1280 x 1024 pixels.

Note: Your Web browser must be JavaScript-enabled in order to access the Web interface.

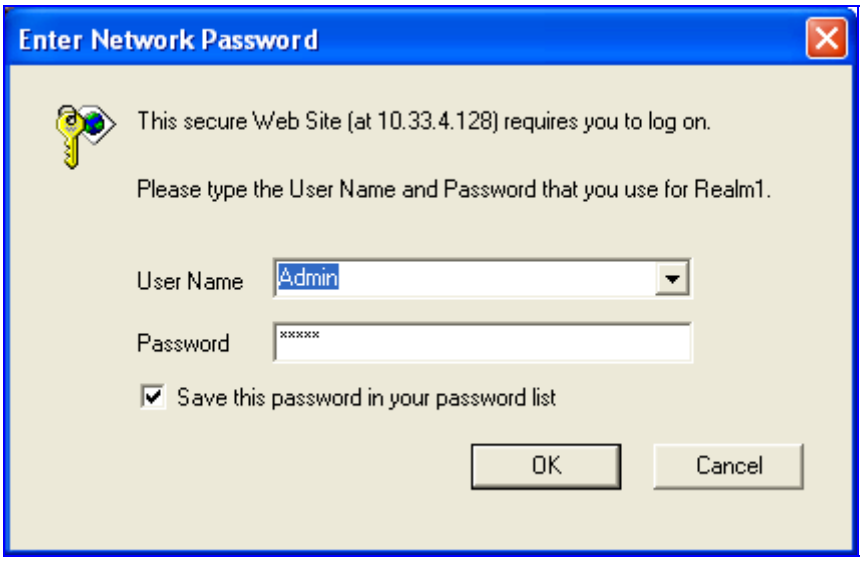
3.2 Accessing the Web Interface

The Web interface can be opened using any standard Web browser (refer to "Computer Requirements" on page 25). When initially accessing the Web interface, use the default user name ('Admin') and password ('Admin'). For changing the login user name and password, refer to "Configuring the Web User Accounts" on page 112).

➢ To access the Web interface, take these 4 steps:

1. Open a standard Web browser application.
2. In the Web browser's Uniform Resource Locator (URL) field, specify the device's IP address (e.g., http://10.1.10.10); the Web interface's 'Enter Network Password' dialog box appears, as shown in the figure below:

Figure 3-1: Enter Network Password Screen


Enter Network Password
This secure Web Site (at 10.33.4.128) requires you to log on.
Please type the User Name and Password that you use for Realm1.
User Name
Admin
Password
*****
Save this password in your password list
OK
Cancel

3. In the 'User Name' and 'Password' fields, enter the case-sensitive, user name and password.
4. Click the **OK** button; the Web interface is accessed, displaying the 'Home' page (for a detailed description of the 'Home' page, refer to "Using the Home Page" on page 52).

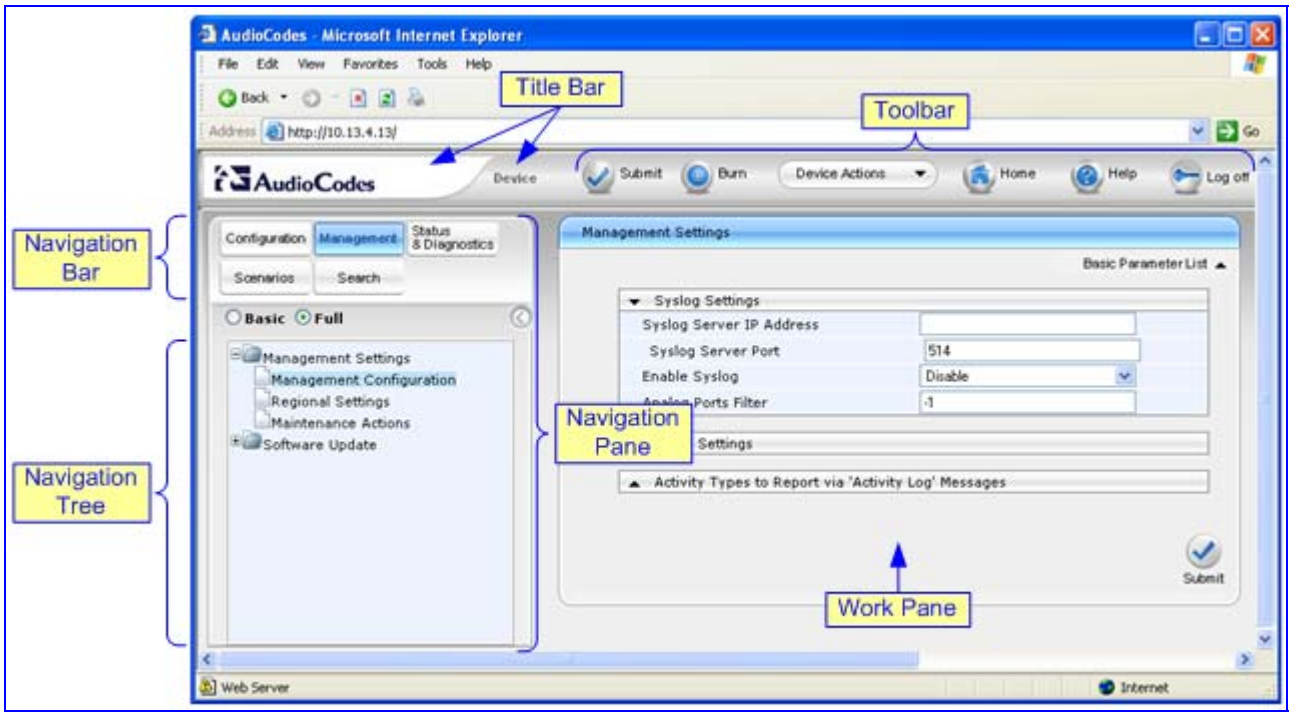
Note: If access to the device's Web interface is denied ("Unauthorized") due to Microsoft Internet Explorer security settings, perform the following troubleshooting procedures:

1. Delete all cookies in the Temporary Internet Files folder. If this does not resolve the problem, the security settings may need to be altered (continue with Step 2).
2. In Internet Explorer, navigate to **Tools** menu > **Internet Options** > **Security** tab > **Custom Level**, and then scroll down to the Logon options and select **Prompt for username and password**. Select the **Advanced** tab, and then scroll down until the HTTP 1.1 Settings are displayed and verify that **Use HTTP 1.1** is selected.
3. Quit and start the Web browser again.

3.3 Getting Acquainted with the Web Interface

The figure below displays the general layout of the Graphical User Interface (GUI) of the Web interface:

Figure 3-2: Main Areas of the Web Interface GUI



The Web GUI is composed of the following main areas:

- **Title bar:** Displays the corporate logo and product name. For replacing the logo with another image or text, refer to "Replacing the Corporate Logo" on page 48. For customizing the product name, refer to "Customizing the Product Name" on page 50.
- **Toolbar:** Provides frequently required command buttons for configuration (refer to "Toolbar" on page 28).
- **Navigation Pane:** Consists of the following areas:
  - **Navigation bar:** Provides tabs for accessing the configuration menus (refer to "Navigation Tree" on page 29), creating a Scenario (refer to "Scenarios" on page 40), and searching *ini* file parameters that have corresponding Web interface parameters (refer to "Searching for Configuration Parameters" on page 38).
  - **Navigation tree:** Displays the elements pertaining to the tab selected on the Navigation bar (tree-like structure of the configuration menus, Scenario Steps, or Search engine) .
- **Work pane:** Displays configuration pages where all configuration is performed (refer to "Working with Configuration Pages" on page 31).

3.3.1 Toolbar

The toolbar provides command buttons for quick-and-easy access to frequently required commands, as described in the table below:

Table 3-1: Description of Toolbar Buttons

Icon	Button Name	Description
	Submit	Applies parameter settings to the device (refer to "Saving Configuration" on page 265). **Note:** This icon is grayed out when not applicable to the currently opened page.
	Burn	Saves parameter settings to flash memory (refer to "Saving Configuration" on page 265).
--	**Device Actions**	Opens a drop-down menu list with frequently needed commands: ▪ **Load Configuration File:** opens the 'Configuration File' page for loading an *ini* file (refer to "Backing Up and Restoring Configuration" on page 276). ▪ **Save Configuration File:** opens the 'Configuration File' page for saving the *ini* file to a PC (refer to "Backing Up and Restoring Configuration" on page 276). ▪ **Reset:** opens the 'Maintenance Actions' page for resetting the device (refer to "Resetting the Device" on page 262). ▪ **Software Upgrade Wizard:** opens the 'Software Upgrade Wizard' page for upgrading the device's software (refer to "Software Upgrade Wizard" on page 272).
	Home	Opens the 'Home' page (refer to "Using the Home Page" on page 52).
	Help	Opens the Online Help topic of the currently opened configuration page in the Work pane (refer to "Getting Help" on page 52).
	Log off	Logs off a session with the Web interface (refer to "Logging Off the Web Interface" on page 62).

Note: If you modify parameters that take effect only after a device reset, after you click the **Submit** button, the toolbar displays the word "Reset" (in red color), as shown in the figure below. This is a reminder to later save ('burn') your settings to flash memory and reset the device.

Figure 3-3: "Reset" Displayed on Toolbar

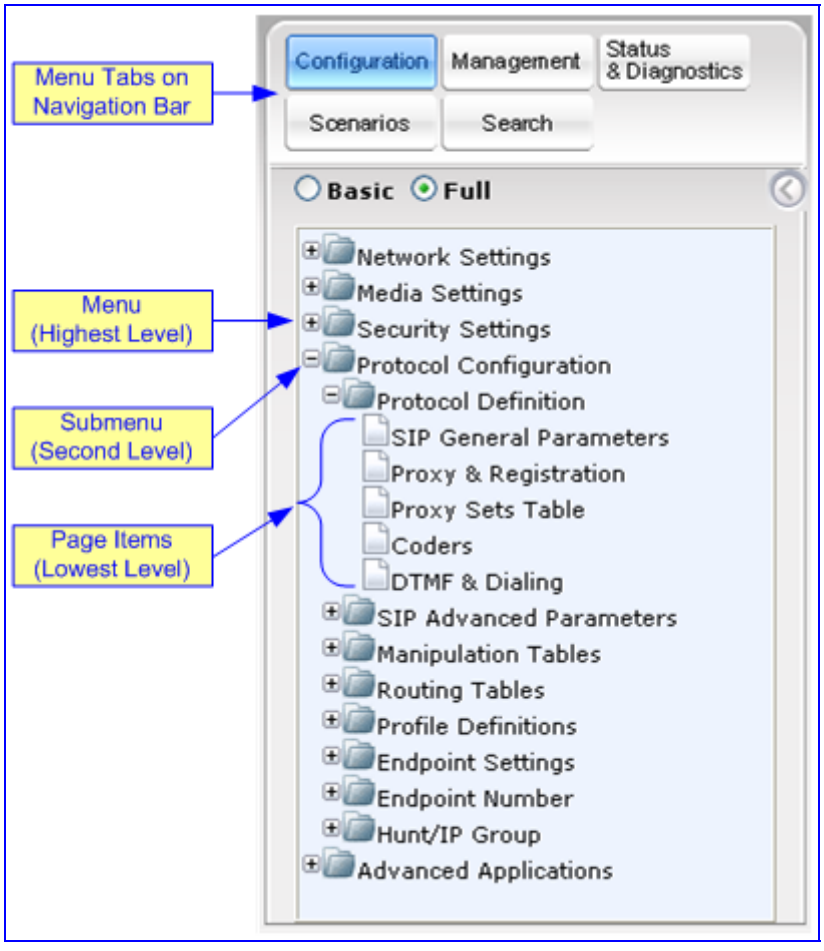
3.3.2 Navigation Tree

The Navigation tree, located in the Navigation pane, displays the menus (pertaining to the menu tab selected on the Navigation bar) used for accessing the configuration pages. The Navigation tree displays a tree-like structure of menus. You can easily drill-down to the required page item level to open its corresponding page in the Work pane.

The terminology used throughout this manual for referring to the hierarchical structure of the tree is as follows:

- *menu*: first level (highest level)
- *submenu*: second level - contained within a menu.
- *page item*: last level (lowest level in a menu) - contained within a menu or submenu.

Figure 3-4: Terminology for Navigation Tree Levels



➢ To view menus in the Navigation tree, take this step:

- On the Navigation bar, select the required tab:
  - **Configuration** (refer to "Configuration Tab" on page 64)
  - **Management** (refer to "Management Tab" on page 253)
  - **Status & Diagnostics** (refer to "Status & Diagnostics Tab" on page 278)

➢ To navigate to a page, take these 2 steps:

1. Navigate to the required page item, by performing the following:
   - Drilling-down using the **plus** ⊞ signs to expand the menus and submenus
   - Drilling-up using the **minus** ⊟ signs to collapse the menus and submenus
2. Select the required page item; the page opens in the Work pane.

3.3.2.1 Displaying Navigation Tree in Basic and Full View

You can view an expanded or reduced Navigation tree display regarding the number of listed menus and submenus. This is relevant when using the configuration tabs (**Configuration**, **Management**, and **Status & Diagnostics**) on the Navigation bar.

The Navigation tree menu can be displayed in one of two views:

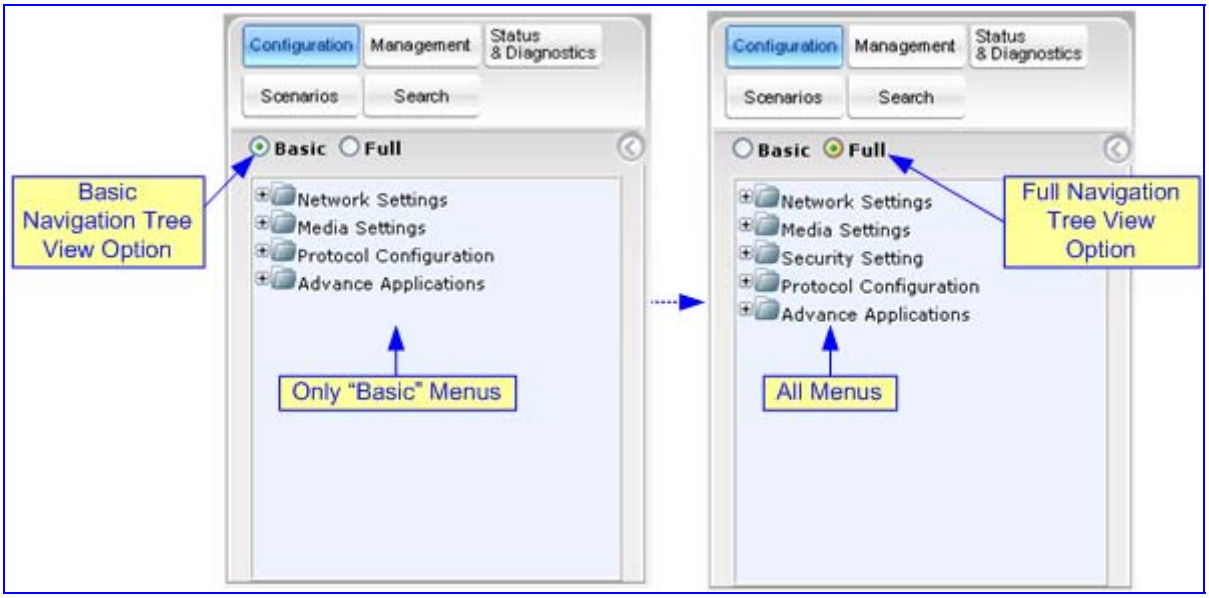
- **Basic:** displays only commonly used menus
- **Full:** displays all the menus pertaining to a configuration tab.

The advantage of the Basic view is that it prevents "cluttering" the Navigation tree with menus that may not be required. Therefore, a Basic view allows you to easily locate required menus.

➢ To toggle between Full and Basic view, take this step:

- Select the **Basic** option (located below the Navigation bar) to display a reduced menu tree; select the **Full** option to display all the menus. By default, the **Basic** option is selected.

Figure 3-5: Navigation Tree in Basic and Full View



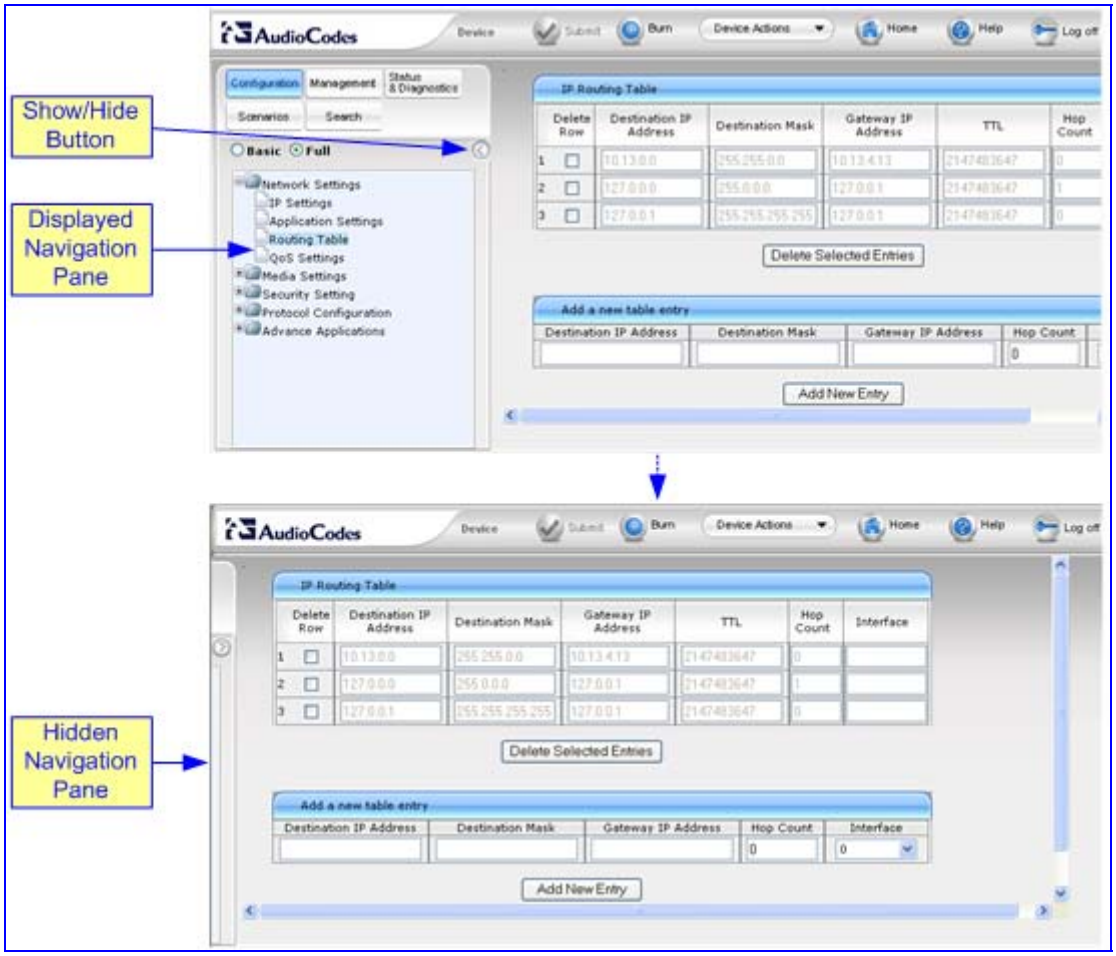
Note: When in Scenario mode (refer to "Scenarios" on page 40), the Navigation tree is displayed in 'Full' view (i.e., all menus are displayed in the Navigation tree).

3.3.2.2 Showing / Hiding the Navigation Pane

The Navigation pane can be hidden to provide more space for elements displayed in the Work pane. This is especially useful when the Work pane displays a page with a table that's wider than the Work pane and to view the all the columns, you need to use scroll bars. The arrow button located just below the Navigation bar is used to hide and show the Navigation pane.

- **To hide the Navigation pane:** click the left-pointing arrow ; the pane is hidden and the button is replaced by the right-pointing arrow button.
- **To show the Navigation pane:** click the right-pointing arrow ; the pane is displayed and the button is replaced by the left-pointing arrow button.

Figure 3-6: Showing and Hiding Navigation Pane



3.3.3 Working with Configuration Pages

The configuration pages contain the parameters for configuring the device. The configuration pages are displayed in the Work pane, which is located to the right of the Navigation pane.

3.3.3.1 Accessing Pages

The configuration pages are accessed by clicking the required page item in the Navigation tree.

➢ To open a configuration page in the Work pane, take these 2 steps:

1. On the Navigation bar, click the required tab:
   - **Configuration** (refer to "Configuration Tab" on page 64)
   - **Management** (refer to "Management Tab" on page 253)
   - **Status & Diagnostics** (refer to "Status & Diagnostics Tab" on page 278)

   The menus of the selected tab appears in the Navigation tree.
2. In the Navigation tree, drill-down to the required page item; the page opens in the Work pane.

You can also access previously opened pages, by clicking your Web browser's **Back** button until you have reached the required page. This is useful if you want to view pages in which you have performed configurations in the current Web session.

Notes:

- You can also access certain pages from the **Device Actions** button located on the toolbar (refer to "Toolbar" on page 28).
- To view all the menus in the Navigation tree, ensure that the Navigation tree is in 'Full' view (refer to "Displaying Navigation Tree in Basic and Full View" on page 30).
- To get Online Help for the currently opened page, refer to "Getting Help" on page 52.
- Certain pages may not be accessible if your Web user account's access level is low (refer to "Configuring the Web User Accounts" on page 112).

3.3.3.2 Viewing Parameters

For convenience, some pages allow you to view a reduced or expanded display of parameters. A reduced display allows you to easily identify required parameters, enabling you to quickly configure your device.

The Web interface provides you with two methods for handling the display of page parameters:

- Display of "basic" and "advanced" parameters (refer to "Displaying Basic and Advanced Parameters" on page 33)
- Display of parameter groups (refer to "Showing / Hiding Parameter Groups" on page 34)

Note: Certain pages may only be read-only if your Web user account's access level is low (refer to "Configuring the Web User Accounts" on page 112). If a page is read-only, 'Read-Only Mode' is displayed at the bottom of the page.

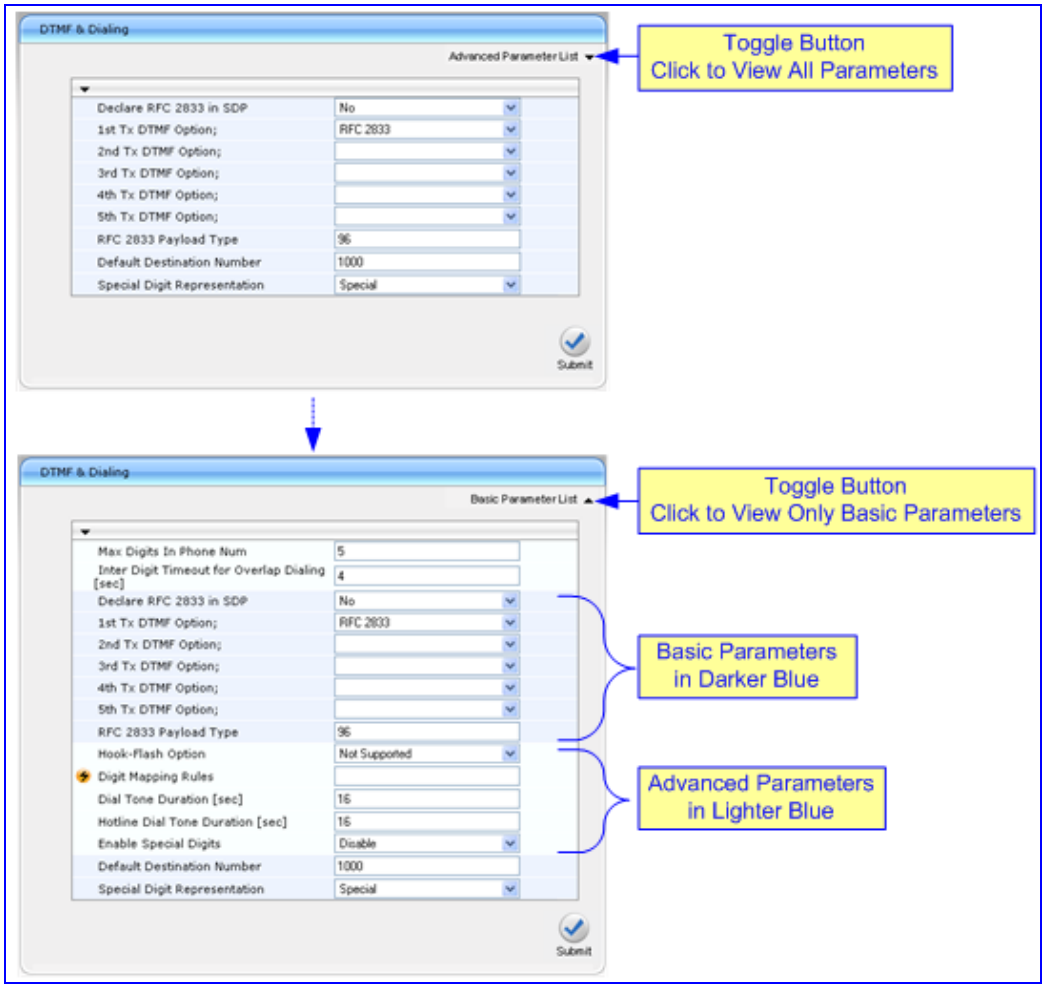
3.3.3.2.1 Displaying Basic and Advanced Parameters

Some pages provide you with an **Advanced Parameter List** / **Basic Parameter List** toggle button that allows you to show or hide advanced parameters (in addition to displaying the basic parameters). This button is located on the top-right corner of the page and has two states:

- **Advanced Parameter List** button with down-pointing arrow: click this button to display all parameters.
- **Basic Parameter List** button with up-pointing arrow: click this button to show only common (*basic*) parameters.

The figure below shows an example of a page displaying basic parameters only, and then showing advanced parameters as well, using the **Advanced Parameter List** button.

Figure 3-7: Toggling between Basic and Advanced Page View



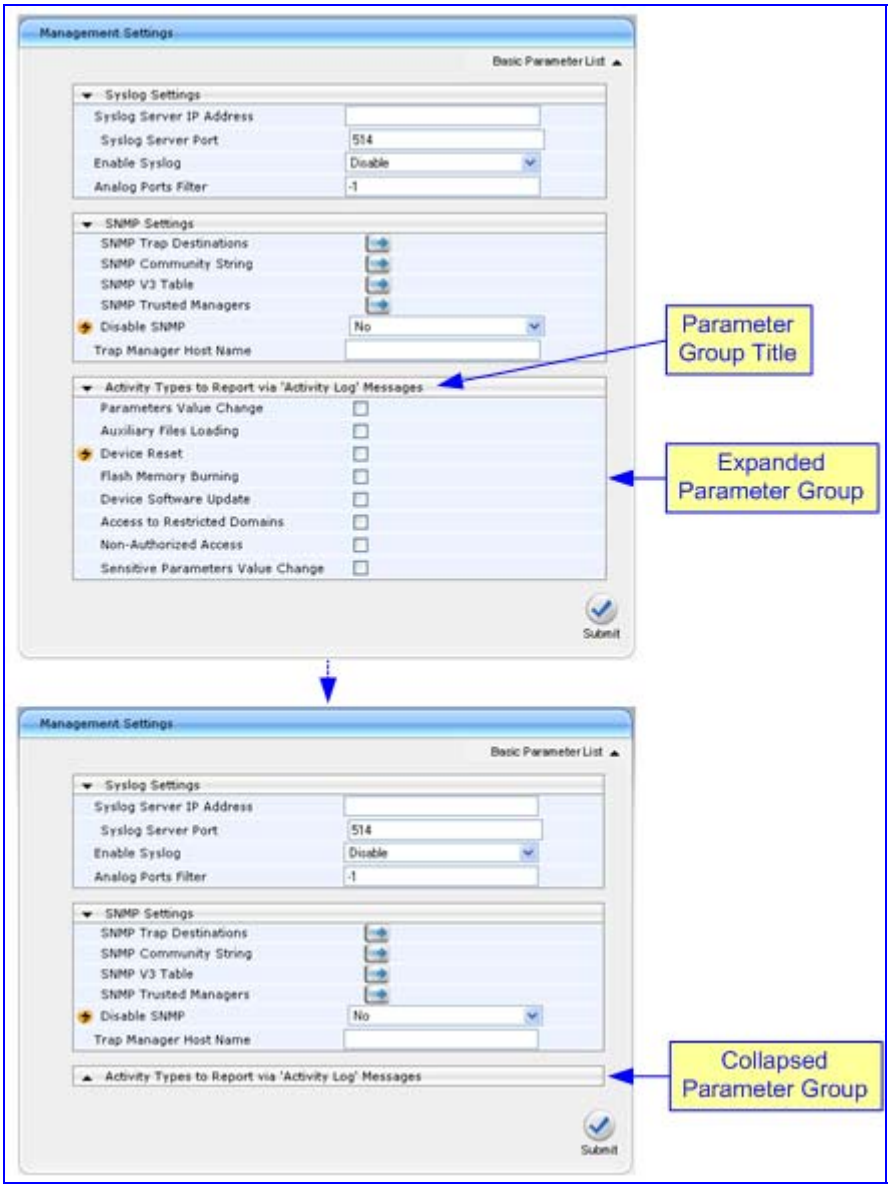
For ease of identification, the basic parameters are displayed with a darker blue color background than the advanced parameters.

Note: When the Navigation tree is in 'Full' mode (refer to "Navigation Tree" on page 29), configuration pages display all their parameters (i.e., the 'Advanced Parameter List' view is displayed).

3.3.3.2.2 Showing / Hiding Parameter Groups

Some pages provide groups of parameters, which can be hidden or shown. To toggle between hiding and showing a group, simply click the group name button that appears above each group. The button appears with a down-pointing or up-pointing arrow, indicating that it can be collapsed or expanded when clicked, respectively.

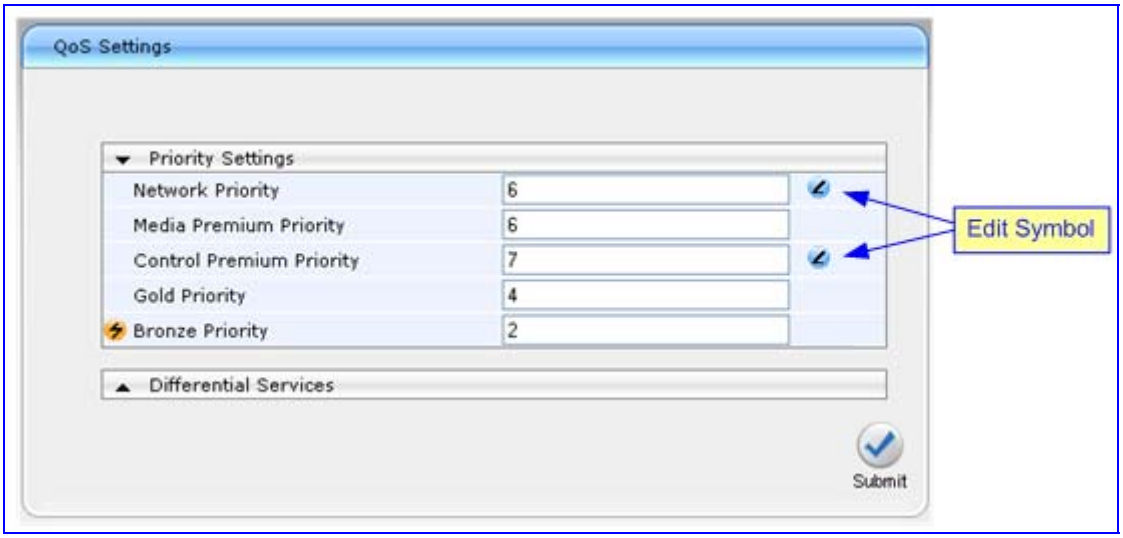
Figure 3-8: Expanding and Collapsing Parameter Groups



3.3.3.3 Modifying and Saving Parameters

When you change parameter values on a page, the **Edit** symbol appears to the right of these parameters. This is especially useful for indicating the parameters that you have currently modified (before applying the changes). After you save your parameter modifications (refer to the procedure described below), the **Edit** symbols disappear.

Figure 3-9: Editing Symbol after Modifying Parameter Value



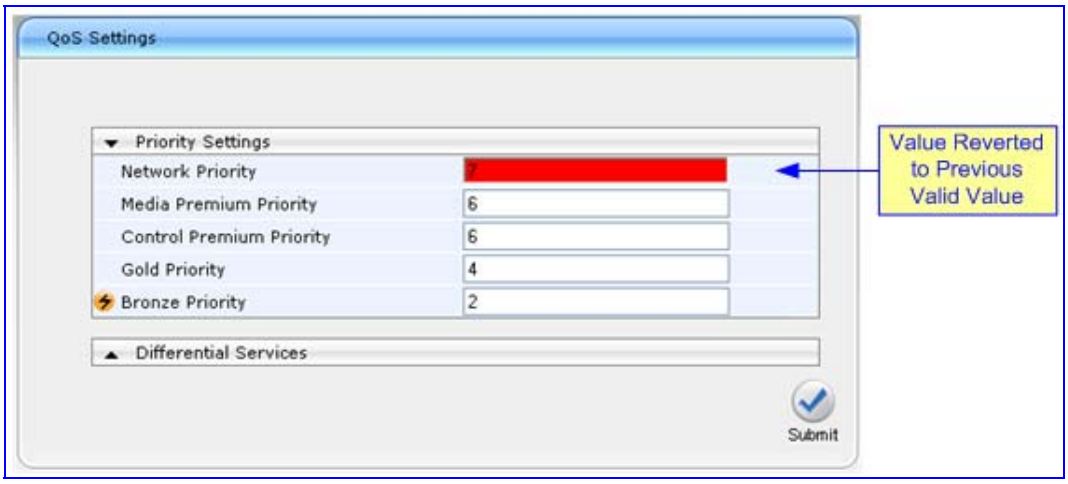
➢ To save configuration changes on a page to the device's volatile memory (RAM), take this step:

- Click the **Submit** button, which is located near the bottom of the page in which you are working; modifications to parameters with on-the-fly capabilities are immediately applied to the device and take effect; other parameters (displayed on the page with the lightning symbol) are not changeable on-the-fly and require a device reset (refer to "Resetting the Device" on page 262) before taking effect.

Note: Parameters saved to the volatile memory (by clicking **Submit**), revert to their previous settings after a hardware or software reset (or if the device is powered down). Therefore, to ensure parameter changes (whether on-the-fly or not) are retained, you need to save ('burn') them to the device's non-volatile memory, i.e., flash (refer to "Saving Configuration" on page 265).

If you enter an invalid parameter value (e.g., not in the range of permitted values) and then click **Submit**, a message box appears notifying you of the invalid value. In addition, the parameter value reverts to its previous value and is highlighted in red, as shown in the figure below:

Figure 3-10: Value Reverts to Previous Valid Value



3.3.3.4 Entering Phone Numbers in Various Tables

Phone numbers or prefixes that you enter in various tables throughout the Web interface such as the 'Tel to IP Routing' table, must only be entered as digits without any other characters. For example, if you wish to enter the phone number 555-1212, it must be entered as 5551212 without the hyphen (-). If the hyphen is entered, the entry is invalid.

3.3.3.5 Working with Tables

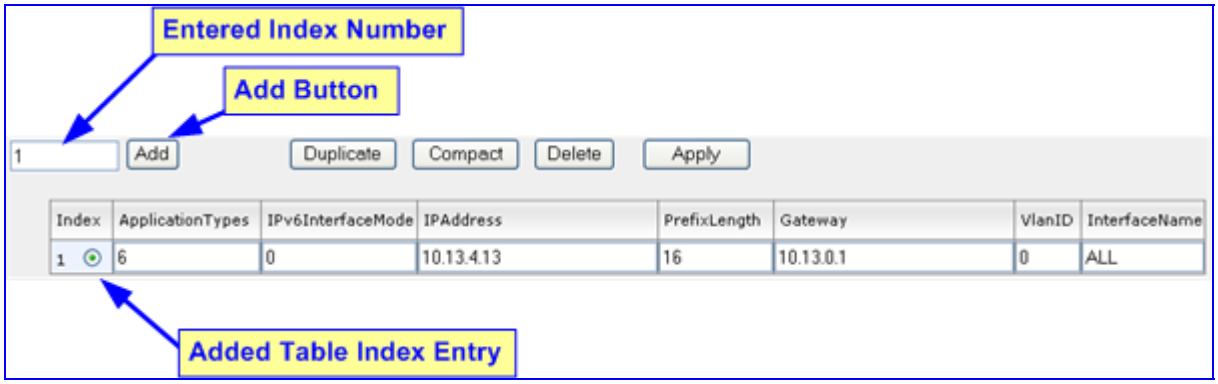
The Web interface includes many configuration pages that provide tables for configuring the device. Some of these tables provide the following command buttons:

- **Add:** adds an index entry to the table.
- **Duplicate:** duplicates a selected, existing index entry.
- **Compact:** organizes the index entries in ascending, consecutive order.
- **Delete:** deletes a selected index entry.
- **Apply:** saves the configuration.

➢ To add an entry to a table, take these 2 steps:

1. In the 'Add' field, enter the desired index entry number, and then click **Add**; an index entry row appears in the table:

Figure 3-11: Adding an Index Entry to a Table



2. Click **Apply** to save the index entry.

Notes:

- Before you can add another index entry, you must ensure that you have applied the previously added index entry (by clicking **Apply**).
- If you leave the 'Add' field blank and then click **Add**, the existing index entries are all incremented by one and the newly added index entry is assigned the index 0.

➢ To add a copy of an existing index table entry, take these 3 steps:

1. In the 'Index' column, select the index that you want to duplicate; the **Edit** button appears.
2. Click **Edit**; the fields in the corresponding index row become available.
3. Click **Duplicate**; a new index entry is added with identical settings as the selected index in Step 1. In addition, all existing index entries are incremented by one and the newly added index entry is assigned the index 0.

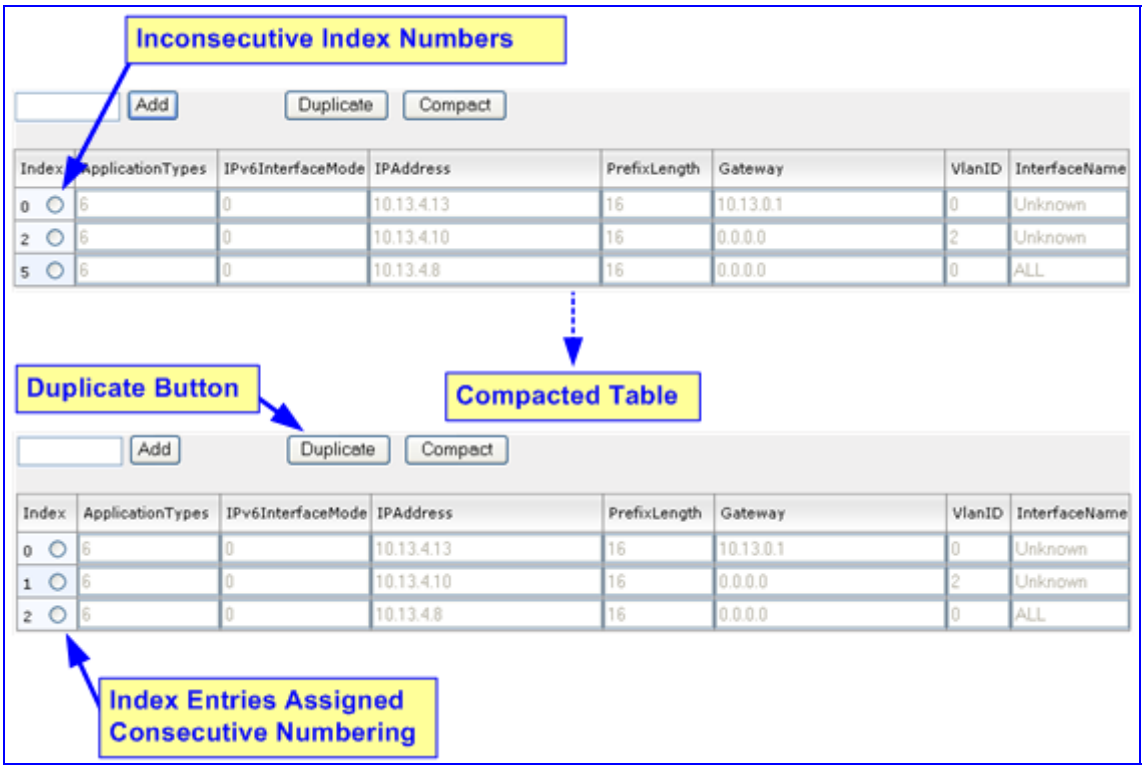
➢ To edit an existing index table entry, take these 3 steps:

1. In the 'Index' column, select the index corresponding to the table row that you want to edit.
2. Click **Edit**; the fields in the corresponding index row become available.
3. Modify the values as required, and then click **Apply**; the new settings are applied.

➢ To organize the index entries in ascending, consecutive order, take the following step:

- Click **Compact**; the index entries are organized in ascending, consecutive order, starting from index 0. For example, if you added three index entries 0, 4, and 6, then the index entry 4 is re-assigned index number 1 and the index entry 6 is re-assigned index number 2.

Figure 3-12: Compacting a Web Interface Table



➢ To delete an existing index table entry, take these 3 steps:

1. In the 'Index' column, select the index corresponding to the table row that you want to delete.
2. Click **Delete**; the table row is removed from the table.

3.3.4 Searching for Configuration Parameters

The Web interface provides a search engine that allows you to search any *ini* file parameter that is configurable by the Web interface (i.e., has a corresponding Web parameter). You can search for a specific parameter (e.g., "EnableIPSec") or a sub-string of that parameter (e.g., "sec"). If you search for a sub-string, all parameters that contain the searched sub-string in their names are listed.

➢ To search for *ini* file parameters configurable in the Web interface, take these 4 steps:

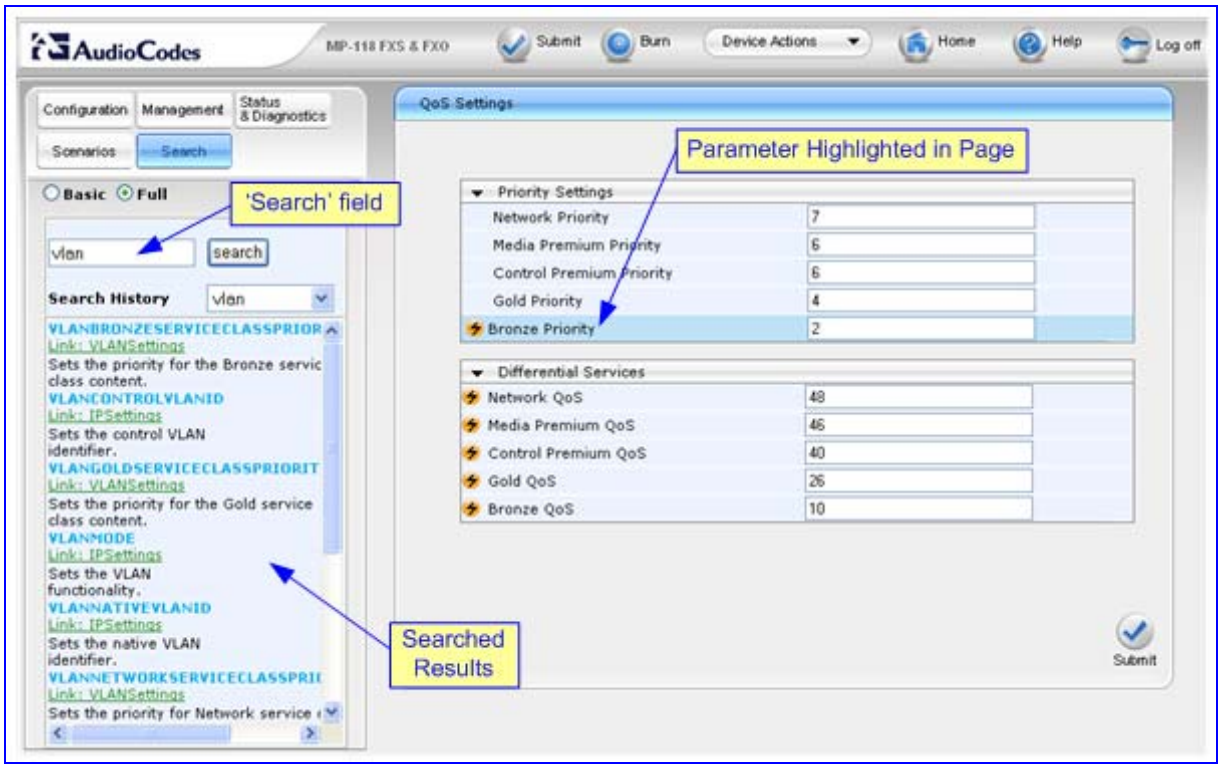
1. On the Navigation bar, click the **Search** tab; the Search engine appears in the Navigation pane.
2. In the 'Search' field, enter the parameter name or sub-string of the parameter name that you want to search. If you have performed a previous search for such a parameter, instead of entering the required string, you can use the 'Search History' drop-down list to select the string (saved from a previous search).
3. Click **Search**; a list of located parameters based on your search appears in the Navigation pane.

   Each searched result displays the following:

   - *ini* file parameter name
   - Link (in green) to its location (page) in the Web interface
   - Brief description of the parameter

4. In the searched list, click the required parameter (link in green) to open the page in which the parameter appears; the relevant page opens in the Work pane and the searched parameter is highlighted for easy identification, as shown in the figure below:

Figure 3-13: Searched Result Screen



Note: If the searched parameter is not located, a notification message is displayed.

3.3.5 Working with Scenarios

The Web interface allows you to create your own "menu" with up to 20 pages selected from the menus in the Navigation tree (i.e., pertaining to the **Configuration**, **Management**, and **Status & Diagnostics** tabs). The "menu" is a set of configuration pages grouped into a logical entity referred to as a *Scenario*. Each page in the Scenario is referred to as a *Step*. For each Step, you can select up to 25 parameters in the page that you want available in the Scenario. Therefore, the Scenario feature is useful in that it allows you quick-and-easy access to commonly used configuration parameters specific to your network environment. When you login to the Web interface, your Scenario is displayed in the Navigation tree, thereby, facilitating your configuration.

Instead of creating a Scenario, you can also load an existing Scenario from a PC to the device (refer to "Loading a Scenario to the Device" on page 46).

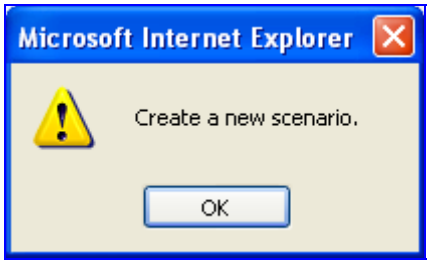
3.3.5.1 Creating a Scenario

The Web interface allows you to create one Scenario with up to 20 configuration pages, as described in the procedure below:

➢ To create a Scenario, take these 10 steps:

1. On the Navigation bar, click the **Scenarios** tab; a message box appears, requesting you to confirm creation of a Scenario:

Figure 3-14: Scenario Creation Confirm Message Box



Note: If a Scenario already exists, the Scenario Loading message box appears.

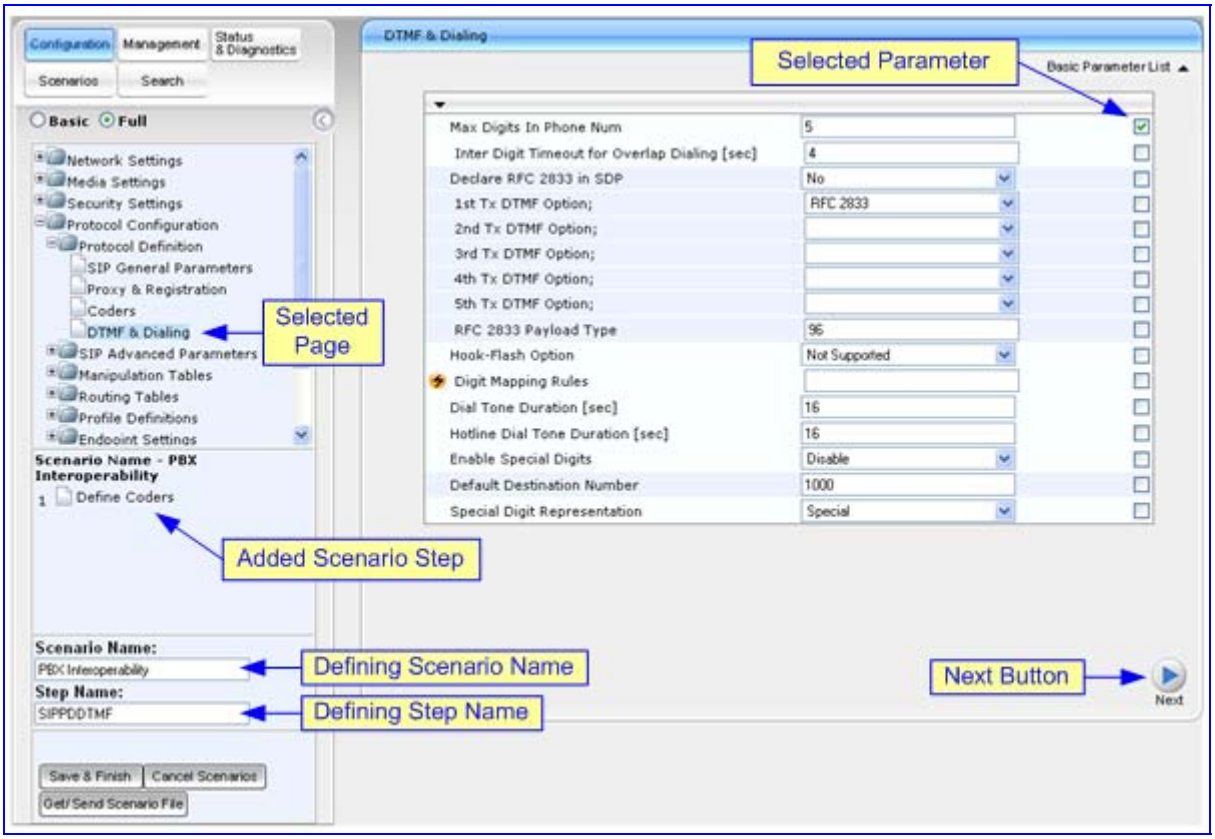
2. Click **OK**; the Scenario mode appears in the Navigation tree as well as the menus of the **Configuration** tab.

   Note: If a Scenario already exists and you wish to create a new one, click the **Create Scenario** button, and then click **OK** in the subsequent message box.

3. In the 'Scenario Name' field, enter an arbitrary name for the Scenario.
4. On the Navigation bar, click the **Configuration** or **Management** tab to display their respective menus in the Navigation tree.
5. In the Navigation tree, select the required page item for the Step, and then in the page itself, select the required parameters by selecting the check boxes corresponding to the parameters.
6. In the 'Step Name' field, enter a name for the Step.

7. Click the **Next** button located at the bottom of the page; the Step is added to the Scenario and appears in the Scenario Step list:

Figure 3-15: Creating a Scenario



8. Repeat steps 5 through 8 to add additional Steps (i.e., pages).

9. When you have added all the required Steps for your Scenario, click the **Save & Finish** button located at the bottom of the Navigation tree; a message box appears informing you that the Scenario has been successfully created.

10. Click **OK**; the Scenario mode is quit and the menu tree of the **Configuration** tab appears in the Navigation tree.

Notes:

- You can add up to 20 Steps to a Scenario, where each Step can contain up to 25 parameters.
- When in Scenario mode, the Navigation tree is in 'Full' display (i.e., all menus are displayed in the Navigation tree) and the configuration pages are in 'Advanced Parameter List' display (i.e., all parameters are shown in the pages). This ensures accessibility to all parameters when creating a Scenario. For a description on the Navigation tree views, refer to "Navigation Tree" on page 29.
- If you previously created a Scenario and you click the **Create Scenario** button, the previously created Scenario is deleted and replaced with the one you are creating.
- Only users with access level of 'Security Administrator' can create a Scenario.

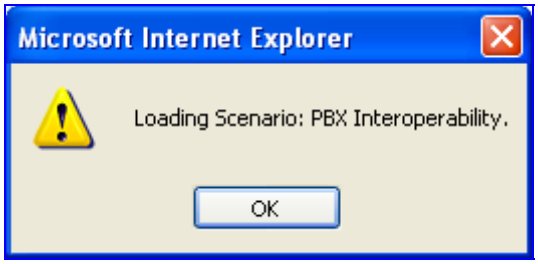
3.3.5.2 Accessing a Scenario

Once you have created the Scenario, you can access it at anytime by following the procedure below:

➢ To access the Scenario, take these 2 steps:

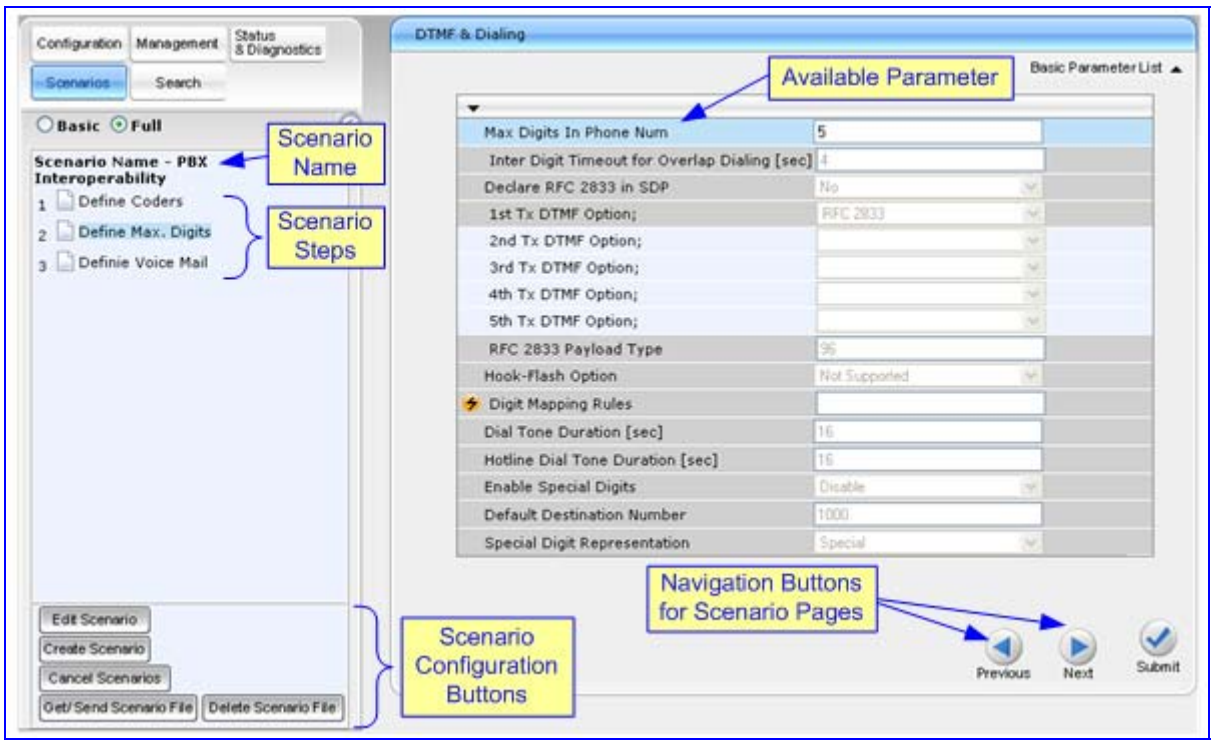
1. On the Navigation bar, select the **Scenario** tab; a message box appears, requesting you to confirm the loading of the Scenario.

Figure 3-16: Scenario Loading Message Box



2. Click **OK**; the Scenario and its Steps appear in the Navigation tree, as shown in the example figure below:

Figure 3-17: Scenario Example



When you select a Scenario Step, the corresponding page is displayed in the Work pane. In each page, the available parameters are indicated by a dark-blue background; the unavailable parameters are indicated by a gray or light-blue background.

To navigate between Scenario Steps, you can perform one of the following:

- In the Navigation tree, click the required Scenario Step.
- In an opened Scenario Step (i.e., page appears in the Work pane), use the following navigation buttons:
  - **Next:** opens the next Step listed in the Scenario.
  - **Previous**: opens the previous Step listed in the Scenario.

Note: If you reset the device while in Scenario mode, after the device resets, you are returned once again to the Scenario mode.

3.3.5.3 Editing a Scenario

You can modify a Scenario anytime by adding or removing Steps (i.e., pages) or parameters, and changing the Scenario name and the Steps' names.

Note: Only users with access level of 'Security Administrator' can edit a Scenario.

➢ To edit a Scenario, take these 6 steps:

1. On the Navigation bar, click the **Scenarios** tab; a message box appears, requesting you to confirm Scenario loading.
2. Click **OK**; the Scenario appears with its Steps in the Navigation tree.
3. Click the **Edit Scenario** button located at the bottom of the Navigation pane; the 'Scenario Name' and 'Step Name' fields appear.
4. You can perform the following edit operations:
   - **Add Steps:**
     a. On the Navigation bar, select the desired tab (i.e., **Configuration** or **Management**); the tab's menu appears in the Navigation tree.
     b. In the Navigation tree, navigate to the desired page item; the corresponding page opens in the Work pane.
     c. In the page, select the required parameter(s) by marking the corresponding check box(es).
     d. Click **Next**.

- **Add or Remove Parameters:**
  a. In the Navigation tree, select the required Step; the corresponding page opens in the Work pane.
  b. To add parameters, select the check boxes corresponding to the desired parameters; to remove parameters, clear the check boxes corresponding to the parameters that you want removed.
  c. Click **Next**.
- **Edit the Step Name:**
  a. In the Navigation tree, select the required Step.
  b. In the 'Step Name' field, modify the Step name.
  c. In the page, click **Next**.
- **Edit the Scenario Name:**
  a. In the 'Scenario Name' field, edit the Scenario name.
  b. In the displayed page, click **Next**.
- **Remove a Step:**
  a. In the Navigation tree, select the required Step; the corresponding page opens in the Work pane.
  b. In the page, clear all the check boxes corresponding to the parameters.
  c. Click **Next**.

5. After clicking **Next**, a message box appears notifying you of the change. Click **OK**.
6. Click **Save & Finish**; a message box appears informing you that the Scenario has been successfully modified. The Scenario mode is exited and the menus of the **Configuration** tab appear in the Navigation tree.

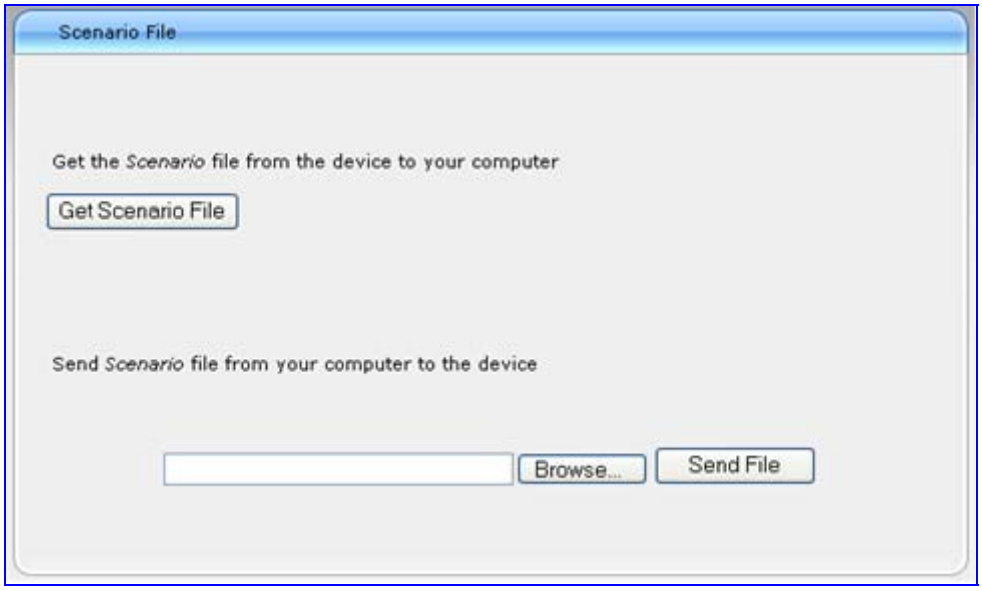
3.3.5.4 Saving a Scenario to a PC

You can save a Scenario to a PC (as a *dat* file). This is especially useful when requiring more than one Scenario to represent different environment setups (e.g., where one includes PBX interoperability and another not). Once you create a Scenario and save it to your PC, you can then keep on saving modifications to it under different Scenario file names. When you require a specific network environment setup, you can simply load the suitable Scenario file from your PC (refer to "Loading a Scenario to the Device" on page 46).

➢ To save a Scenario to a PC, take these 5 steps:

1. On the Navigation bar, click the **Scenarios** tab; the Scenario appears in the Navigation tree.
2. Click the **Get/Send Scenario File** button (located at the bottom of the Navigation tree); the 'Scenario File' page appears, as shown below:

Figure 3-18: Scenario File Page


Scenario File
Get the Scenario file from the device to your computer
Get Scenario File
Send Scenario file from your computer to the device
Browse...
Send File

3. Click the **Get Scenario File** button; the 'File Download' window appears.
4. Click **Save**, and then in the 'Save As' window navigate to the folder to where you want to save the Scenario file. When the file is successfully downloaded to your PC, the 'Download Complete' window appears.
5. Click **Close** to close the 'Download Complete' window.

3.3.5.5 Loading a Scenario to the Device

Instead of creating a Scenario, you can load a Scenario file (*data* file) from your PC to the device.

➢ To load a Scenario to the device, take these 4 steps:

1. On the Navigation bar, click the **Scenarios** tab; the Scenario appears in the Navigation tree.
2. Click the **Get/Send Scenario File** button (located at the bottom of the Navigation tree); the 'Scenario File' page appears (refer to "Saving a Scenario to a PC" on page 44).
3. Click the **Browse** button, and then navigate to the Scenario file stored on your PC.
4. Click the **Send File** button.

Notes:

- You can only load a Scenario file to a device that has an identical hardware configuration setup to the device in which it was created. For example, if the Scenario was created in a device with FXS interfaces, the Scenario cannot be loaded to a device that does not have FXS interfaces.
- The loaded Scenario replaces any existing Scenario.
- You can also load a Scenario file using BootP, by loading an *ini* file that contains the *ini* file parameter ScenarioFileName (refer to "Web and Telnet Parameters" on page 311). The Scenario *dat* file must be located in the same folder as the *ini* file. For a detailed description on BootP, refer to the *Product Reference Manual*.

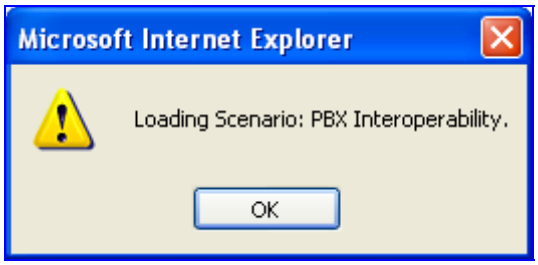
3.3.5.6 Deleting a Scenario

You can delete the Scenario by using the **Delete Scenario File** button, as described in the procedure below:

➢ To delete the Scenario, take these 4 steps:

1. On the Navigation bar, click the **Scenarios** tab; a message box appears, requesting you to confirm:

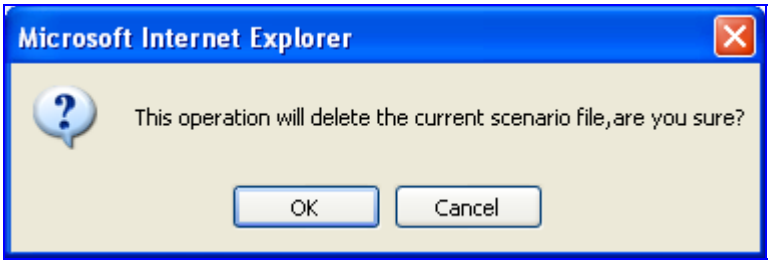
Figure 3-19: Scenario Loading Message Box



2. Click **OK**; the Scenario mode appears in the Navigation tree.

3. Click the **Delete Scenario File** button; a message box appears requesting confirmation for deletion.

Figure 3-20: Message Box for Confirming Scenario Deletion



4. Click **OK**; the Scenario is deleted and the Scenario mode closes.

Note: You can also delete a Scenario using the following alternative methods:

- Loading an empty *dat* file (refer to "Loading a Scenario to the Device" on page 46).
- Loading an *ini* file with the ScenarioFileName parameter set to no value (i.e., ScenarioFileName = "").

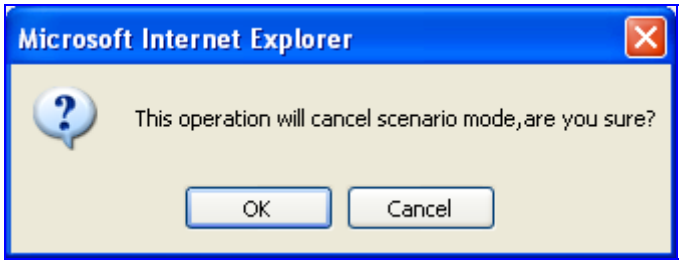
3.3.5.7 Exiting Scenario Mode

When you want to close the Scenario mode after using it for device configuration, follow the procedure below:

➢ To close the Scenario mode, take these 2 steps:

1. Simply click any tab (besides the **Scenarios** tab) on the Navigation bar, or click the **Cancel Scenarios** button located at the bottom of the Navigation tree; a message box appears, requesting you to confirm exiting Scenario mode, as shown below.

Figure 3-21: Confirmation Message Box for Exiting Scenario Mode



2. Click **OK** to exit.

3.3.6 Customizing the Web Interface

You can customize the device's Web interface to suit your company preferences. The following Web interface elements can be customized:

- Corporate logo displayed on the Title bar (refer to "Replacing the Corporate Logo" on page 48)
- Product's name displayed on the Title bar (refer to "Customizing the Product Name" on page 50)
- Login welcome message (refer to "Creating a Login Welcome Message" on page 51)

3.3.6.1 Replacing the Corporate Logo

The corporate logo that appears in the Title bar can be replaced either with a different logo image (refer to "Replacing the Corporate Logo with an Image" on page 48) or text (refer to "Replacing the Corporate Logo with Text" on page 50).

The figure below shows an example of a customized Title bar. The top image displays the Title bar with AudioCodes logo and product name. The bottom image displays a customized Title bar with a different image logo and product name.

Figure 3-22: Customizing Web Logo and Product Name

3.3.6.1.1 Replacing the Corporate Logo with an Image

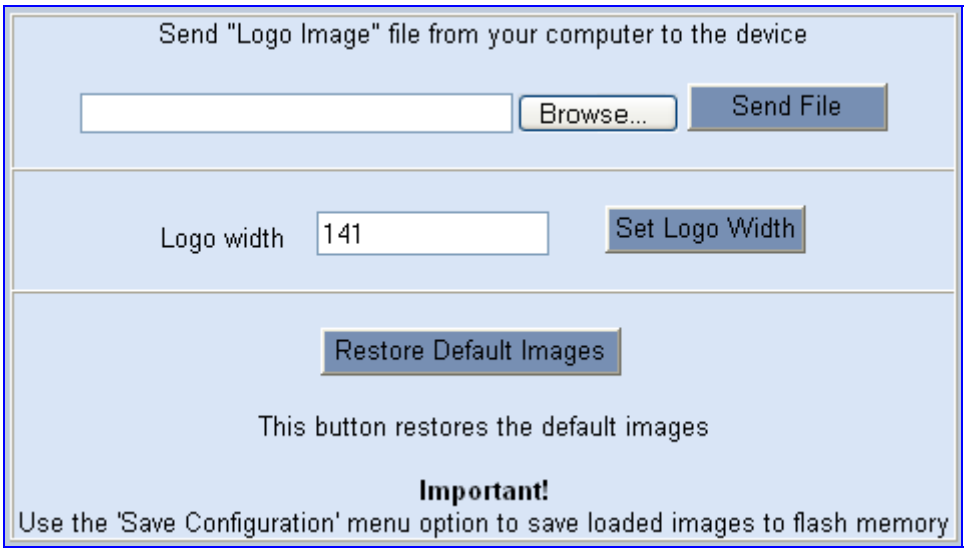
You can replace the logo that appears in the Web interface's Title bar, using either the Web interface or the *ini* file.

➢ **To replace the default logo with a different image via the Web interface, take these 7 steps:**

1. Access the device's Web interface (refer to "Accessing the Web Interface" on page 26).
2. In the URL field, append the case-sensitive suffix 'AdminPage' to the IP address (e.g., http://10.1.229.17/AdminPage); the 'Admin' page appears.

3. On the left pane, click **Image Load to Device**; the 'Image Download' page is displayed, as shown in the figure below:

Figure 3-23: Image Download Screen


Send "Logo Image" file from your computer to the device
Browse...
Send File
Logo width
141
Set Logo Width
Restore Default Images
This button restores the default images
Important!
Use the 'Save Configuration' menu option to save loaded images to flash memory

4. Click the **Browse** button, and then navigate to the folder in which the logo image file that you want to use is located.
5. Click the **Send File** button; the image file uploads to the device. When loading is complete, the page is automatically refreshed and the uploaded logo image is displayed in the Web interface's title bar.
6. If you want to modify the width of the image, in the 'Logo Width' field, enter the new width (in pixels) and then click the **Set Logo Width** button.
7. To save the image to flash memory, refer to "Saving Configuration" on page 265.

Notes:

- The logo image must be a GIF, JPG, or JPEG file.
- The logo image must have a fixed height of 30 pixels. The width can be up to 199 pixels, the default being 141 pixels.
- The size of the image file can be up to 64 Kbytes.

Tip: If you encounter any problem during the loading of the file or you want to restore the default image, click the **Restore Default Images** button.

➢ To replace the default logo with a different image using the *ini* file, take these 3 steps:

1. Place your corporate logo image file on the TFTP server in the same folder where the device's *ini* file is located.
2. Configure the *ini* file parameters as described in the table below. (For a description on using the *ini* file, refer to "Modifying an ini File" on page 296.)
3. Load the *ini* file to the device using BootP / TFTP (i.e., not through the Web interface). For detailed information on the BootP/TFTP application, refer to the *Product Reference Manual*.

Table 3-2: ini File Parameters for Changing Logo Image

Parameter	Description
LogoFileName	The name of the image file for your corporate logo. Use a gif, jpg or jpeg image file. The default is AudioCodes' logo file. **Note:** The length of the name of the image file is limited to 48 characters.
LogoWidth	Width (in pixels) of the logo image. The range is 0 - 199. The default value is 141 (which is the width of AudioCodes' displayed logo). **Note:** The optimal setting depends on the screen resolution settings.

3.3.6.1.2 Replacing the Corporate Logo with Text

The corporate logo can be replaced with a text string instead of an image. To replace AudioCodes' default logo with a text string using the *ini* file, configure the *ini* file parameters listed in the table below. (For a description on using the *ini* file, refer to "Modifying an ini File" on page 296.)

Table 3-3: ini File Parameters for Replacing Logo with Text

Parameter	Description
UseWebLogo	▪ **[0]** = Logo image is used (default). ▪ **[1]** = Text string used instead of a logo image.
WebLogoText	Text string that replaces the logo image. The string can be up to 15 characters.

Note: When a text string is used instead of a logo image, the Web browser's title bar displays the string assigned to the WebLogoText parameter.

3.3.6.2 Customizing the Product Name

You can customize the product name (text) that appears in the Title bar, using the *ini* file parameters listed in the table below. (For a description on using the *ini* file, refer to "Modifying an ini File" on page 296.)

Table 3-4: ini File Parameters for Customizing Product Name

Parameter	Description
UseProductName	Defines whether or not to change the product name: ▪ **[0]** = Don't change the product name (default). ▪ **[1]** = Enable product name change.
UserProductName	The text string that replaces the product name. The default is 'Mediant 1000'. The string can be up to 29 characters.

3.3.6.3 Creating a Login Welcome Message

You can create a Welcome message box (alert message) that appears after each successful login to the device's Web interface. The *ini* file table parameter WelcomeMessage allows you to create the Welcome message. Up to 20 lines of character strings can be defined for the message. If this parameter is not configured, no Welcome message box is displayed after login.

An example of a Welcome message is shown in the figure below:

Figure 3-24: User-Defined Web Welcome Message after Login

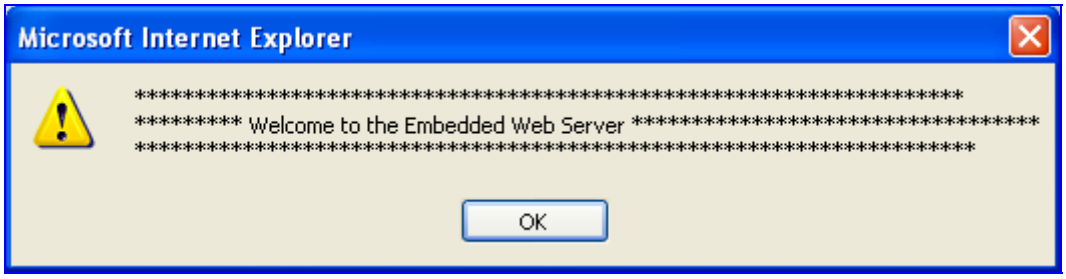


Table 3-5: ini File Parameter for Welcome Login Message

Parameter	Description
WelcomeMessage	Defines the Welcome message that appears after a successful login to the Web interface. The format of this parameter is as follows: [WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text; WelcomeMessage 1 = "..." ; WelcomeMessage 2 = "..." ; WelcomeMessage 3 = "..." ; [\WelcomeMessage] For Example: [WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text; WelcomeMessage 1 = "**********" ; WelcomeMessage 2 = "********** This is a Welcome message ***" ; WelcomeMessage 3 = "**********" ; [\WelcomeMessage] **Note:** Each index represents a line of text in the Welcome message box. Up to 20 indices can be defined.

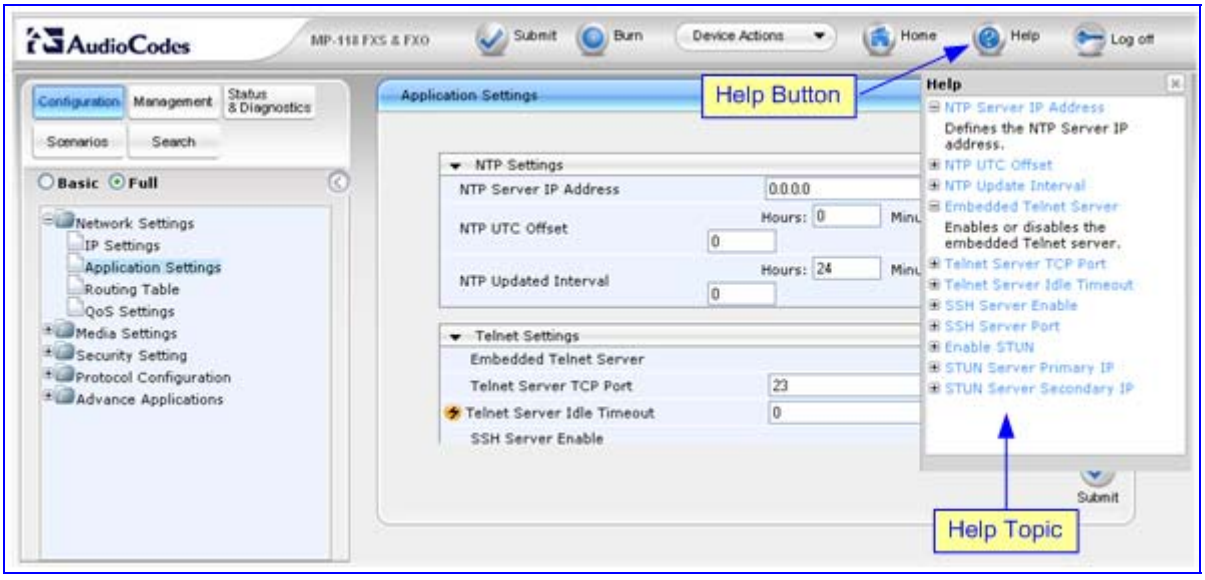
3.3.7 Getting Help

The Web interface provides you with context-sensitive Online Help. The Online Help provides you with brief descriptions of most of the parameters you'll need to successfully configure the device. The Online Help provides descriptions of parameters pertaining to the currently opened page.

➢ To view the Help topic for a currently opened page, take these 4 steps:

1. Using the Navigation tree, open the required page for which you want Help.
2. On the toolbar, click the **Help** button; the Help topic pertaining to the opened page appears, as shown below:

Figure 3-25: Help Topic for Current Page



3. To view a description of a parameter, click the **plus** ⊞ sign to expand the parameter. To collapse the description, click the **minus** ⊟ sign.
4. To close the Help topic, click the **close** button located on the top-right corner of the Help topic window.

Note: Instead of clicking the **Help** button for each page you open, you can open it once for a page, and then simply leave it open. Each time you open a different page, the Help topic pertaining to that page is automatically displayed.

3.3.8 Using the Home Page

The 'Home' page provides you with a graphical display of the device's front panel, displaying color-coded status icons for monitoring the functioning of the device. By default, the 'Home' page is displayed when you access the device's Web interface. When you are configuring the device (in a configuration page), you can always return to the 'Home' page, by simply clicking the **Home** icon on the toolbar. The 'Home' page also displays general device information (in the 'General Information' pane) such as the device's IP address and firmware version.

➢ To access the Home page, take this step:

- On the toolbar, click the **Home** icon; the 'Home' page is displayed:

Figure 3-26: Mediant 1000 Home Page

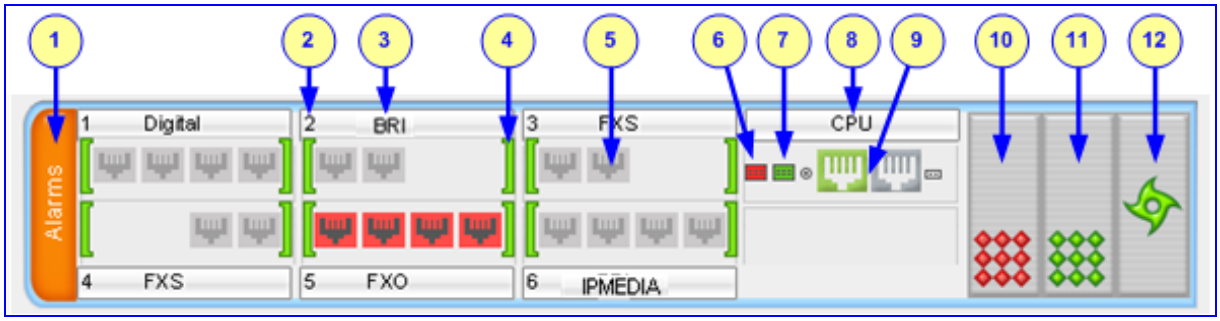
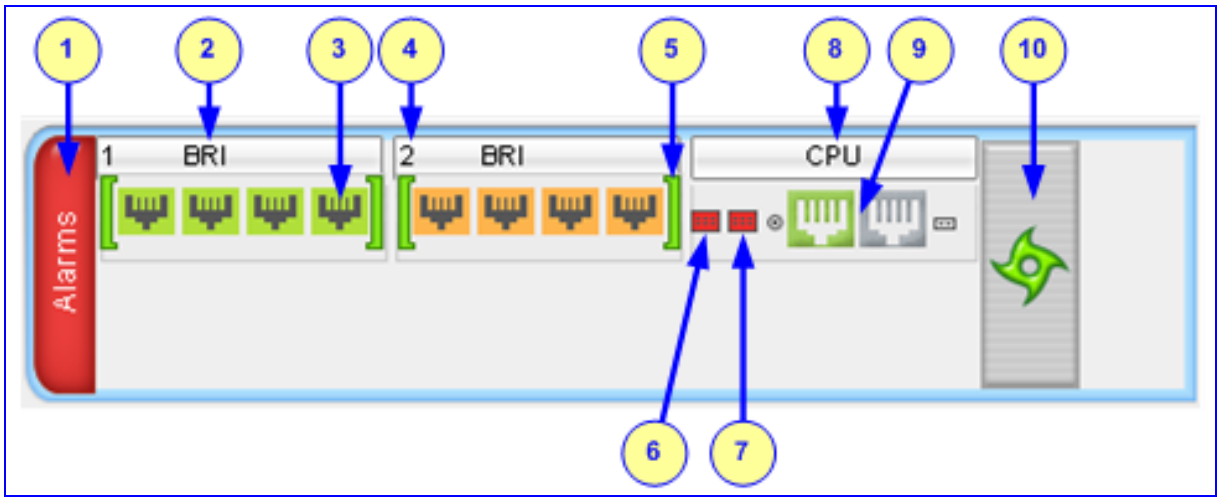


Figure 3-27: Mediant 600 Home Page



Note: The number and type of modules, trunks and channels displayed in the 'Home' page depends on the device's hardware configuration.

The table below describes the areas of the 'Home' page.

Table 3-6: Description of the Areas of the Mediant 1000 Home Page

Item# / Label	Description
1	Displays the highest severity of an active alarm raised (if any) by the device: ▪ Green = No alarms ▪ Red = Critical alarm ▪ Orange = Major alarm ▪ Yellow = Minor alarm To view a list of active alarms in the 'Active Alarms' page (refer to Viewing Active Alarms on page 282), click the Alarms area.
2	Module slot number (1 to 6).
3	Module type (Digital, BRI, IPMEDIA, FXO, or FXS).
4	Module status icon: ▪ (green): Module has been inserted or is correctly configured. ▪ (gray): Module was removed. 'Reserved' is displayed alongside the module's name. ▪ (red): Module failure. 'Failure' is displayed instead of the module's name.

<table>
<tr><th>Item# / Label</th><th>Description</th></tr>
<tr><td>5</td><td>Port (trunk or channel) status icon (refer to Monitoring the Mediant 1000 Trunks and Channels on page 59).
<table>
<tr><th>Icon</th><th>Trunk Description (Digital Module)</th><th>Channel Description (Analog Module)</th></tr>
<tr><td>(grey)</td><td>Disable: Trunk not configured (not in use)</td><td>Inactive: Channel is currently on-hook</td></tr>
<tr><td>(green)</td><td>Active - OK: Trunk synchronized</td><td>Call Connected: Active RTP stream</td></tr>
<tr><td>(yellow)</td><td>RAI Alarm: Remote Alarm Indication (RAI), also known as the Yellow Alarm</td><td>--</td></tr>
<tr><td>(red)</td><td>LOS / LOF Alarm: Loss due to LOS (Loss of Signal) or LOF (Loss of Frame)</td><td>Not Connected: No analog line is connected to this port (FXO only)</td></tr>
<tr><td>(blue)</td><td>AIS Alarm: Alarm Indication Signal (AIS), also known as the Blue Alarm</td><td>Handset Offhook: Channel is off-hook, but there is no active RTP session</td></tr>
<tr><td>(orange)</td><td>D-Channel Alarm: D-channel alarm</td><td>--</td></tr>
</table></td></tr>
<tr><td>6</td><td>Dry Contact (normally open) status icon
▪ (green): Dry Contact is open (normal).
▪ (red): Dry contact is closed.</td></tr>
<tr><td>7</td><td>Dry Contact (normally closed) status icon:
▪ (green): Dry Contact is closed (normal).
▪ (red): Dry contact is open.</td></tr>
<tr><td>8</td><td>CPU module.</td></tr>
<tr><td>9</td><td>Ethernet port status icons:
▪ (green): Ethernet link is working.
▪ (gray): Ethernet link not configured.
You can also view detailed Ethernet port information in the 'Ethernet Port Information' page (refer to Viewing Ethernet Port Information on page 280), by clicking the icon.</td></tr>
<tr><td>10</td><td>Power Supply Unit 1 status icon:
▪ (green): Power supply is operating.
▪ (red): Power supply failure or no power supply unit installed.</td></tr>
<tr><td>11</td><td>Power Supply Unit 2 status indicator. Refer to Item #10 for an explanation.</td></tr>
<tr><td>12</td><td>Fan tray unit status icon:
▪ (green): Fan tray operating.
▪ (red): Fan tray failure.</td></tr>
</table>

Table 3-7: Description of the Areas of the Mediant 600 Home Page

Item# / Label	Description
1	Displays the highest severity of an active alarm raised (if any) by the device: ▪ Green = No alarms ▪ Red = Critical alarm ▪ Orange = Major alarm ▪ Yellow = Minor alarm To view a list of active alarms in the 'Active Alarms' page (refer to Viewing Active Alarms on page 282), click the Alarms area.
2	Module type (Digital or BRI).
3	Module's port status icon: ▪ (gray): Disable - Trunk not configured (not in use). ▪ (green): Active OK - Trunk synchronized. ▪ (yellow): RAI Alarm - Remote Alarm Indication (RAI) also known as the 'Yellow' Alarm. ▪ (red): LOS / LOF Alarm - loss due to Loss of Signal (LOS) or Loss of Frame (LOF). ▪ (blue): AIS Alarm - Alarm Indication Signal (AIS) also known as the 'Blue' Alarm. ▪ (orange): D-Channel Alarm - D-channel alarm. You can also view the channel's port settings (refer to Viewing Port Information on page 58) and assign a name to the port (refer to Assigning a Name to a Port on page 57).
4	Module slot number (1 to 2).
5	Module status icon: ▪ (green): Module has been inserted or is correctly configured. ▪ (gray): Module was removed - 'Reserved' is displayed alongside the module's name. ▪ (red): Module failure - 'Failure' is displayed instead of the module's name.
6	Dry Contact (normally open - Critical alarm) status icons: ▪ (green): Dry Contact is open (normal). ▪ (red): Dry contact is closed.
7	Dry Contact (normally closed - Major alarm) status icons: ▪ (green): Dry Contact is closed (normal). ▪ (red): Dry contact is open.
8	CPU module.

Item# / Label	Description
9	Ethernet port status icons: ▪ (green): Ethernet link is working. ▪ (gray): Ethernet link not configured. You can also view detailed Ethernet port information in the 'Ethernet Port Information' page (refer to Viewing Ethernet Port Information on page 280), by clicking this icon.
10	Fan tray unit status icon: ▪ (green): Fan tray operating. ▪ Red: Fan tray failure.

3.3.8.1 Assigning a Name to a Port

The 'Home' page allows you to assign an arbitrary name or a brief description to each port. This description appears as a tooltip when you move your mouse over the port.

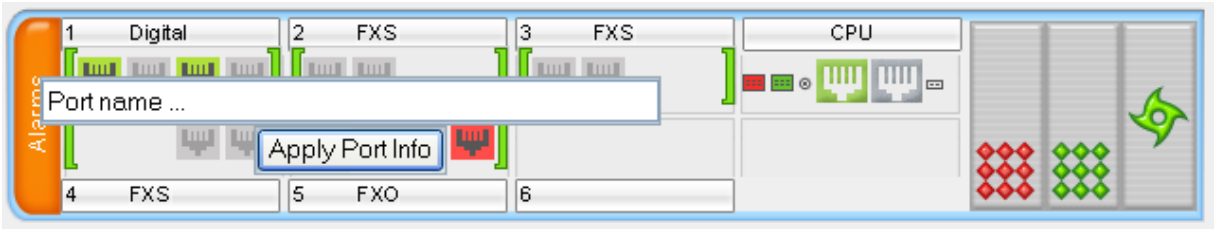
➢ To add a port description, take these 3 steps:

1. Click the required port icon; a shortcut menu appears, as shown below:

Figure 3-28: Shortcut Menu (Example, Mediant 1000 Home Page)

2. From the shortcut menu, choose **Update Port Info**; a text box appears.

Figure 3-29: Text Box for Typing Port Name (Example, Mediant 1000 Home Page)



3. Type a brief description for the port, and then click **Apply Port Info**.

3.3.8.2 Viewing Analog Port Information

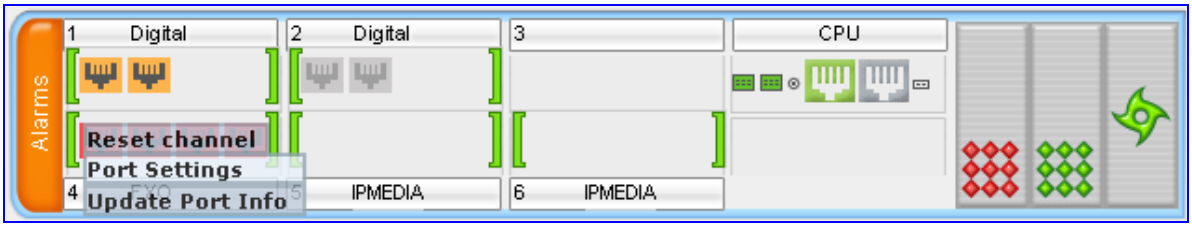
The 'Home' page allows you to view detailed information on a specific FXS or FXO (analog) port such as RTP/RTCP and voice settings.

Note: This section is applicable only to Mediant 1000.

➢ **To view detailed port information, take these 3 steps:**

1. Click the port for which you want to view port settings; the shortcut menu appears.

Figure 3-30: Selecting Port Settings from Shortcut Menu



2. From the shortcut menu, click **Port Settings**; the 'Basic Channel Information' screen appears.

Figure 3-31: Basic Information Screen

SIP ◆ Basic ◆ RTP/RTCP ◆ Voice Settings

Channel Identifier:	6
Status:	Inactive
Call ID:	0
Endpoint ID:	
Call Duration [sec]:	0
Call Type:	Voice
Call Destination:	10.8.230.155
Coder:	G711Alaw_64
Last Current Disconnect Duration:	0
Line Current[mA]:	0
Line Voltage[V]:	0
Hook(0-Onhook, 1-Off hook):	0
Ring(0-Off, 1-On):	0
Line Connected(0-Disconnected, 1-Connected):	0
Polarity state(0-Normal, 1-Reversed, 2-N\A):	0
Line polarity(0-Positive, 1-Negative):	0
Message Waiting Indication(0-Off, 1-On):	0

3. To view RTP/RTCP or voice settings, click the relevant button.

3.3.8.3 Resetting an Analog Channel

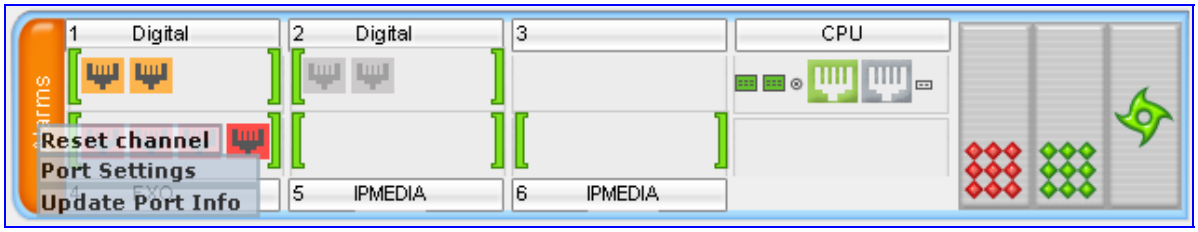
The 'Home' page allows you to inactivate (*reset*) an FXO or FXS analog channel. This is sometimes useful in scenarios, for example, when the device (FXO) is connected to a PBX and the communication between the two can't be disconnected (e.g., when using reverse polarity).

Note: This section is applicable only to Mediant 1000.

➢ To reset a channel, take this step:

- Click the required FXS or FXO port icon, and then from the shortcut menu, choose **Reset Channel**; the channel is changed to inactive (i.e., the port icon is displayed in grey).

Figure 3-32: Resetting Channel



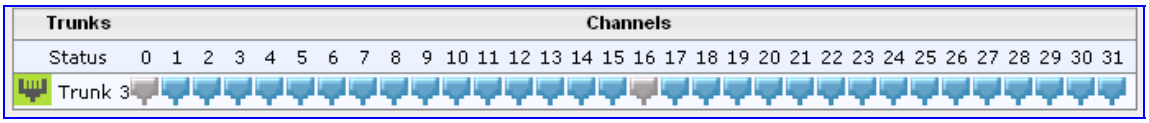
3.3.8.4 Viewing Trunks' Channels

The 'Home' page allows you to drill-down to view a detailed status of the channels pertaining to a trunk In addition, you can also view the trunk's configuration.

➢ To view a detailed status of a trunk's channels, take these 5 steps:

1. In the Home page, click the desired trunk of whose status you want to view; a shortcut menu appears.
2. From the shortcut menu, choose **Port Settings**; the 'Trunks & Channels Status' page pertaining to the specific trunk appears:

Figure 3-33: Trunks & Channels Status Page (Example, Mediant 1000)



The color-coding for the status of the trunk's channels status is described in the table below:

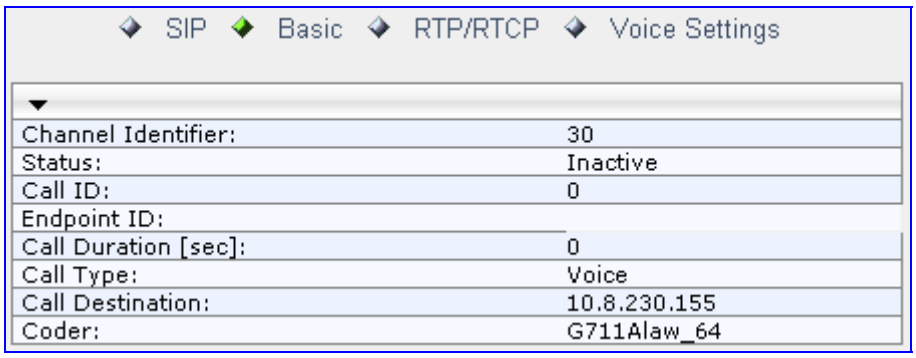
Table 3-8: Color-Coding Status for Trunk's Channels

Icon	Color	Label	Description
	Light blue	**Inactive**	Configured, but currently no call
	Green	**Active**	Call in progress (RTP traffic)
	Purple	**SS7**	Configured for SS7 (Currently not supported)
	Grey	**Non Voice**	Not configured
	Blue	**ISDN Signaling**	Configured as a D-channel
	Yellow	**CAS Blocked**	--

3. To view the configuration settings of the trunk and / or to modify the trunk's settings, in the 'Trunks & Channels Status' screen, click the **Trunk** icon, and then from the shortcut menu, choose **Port Settings**; The 'Trunk Settings' screen appears. (For detailed information on configuring the trunk in this screen, refer to "Configuring the Trunk Settings" on page 96.)

4. To view information of a specific trunk's channel, in the 'Trunks & Channels Status' screen, click the required **Channel** icon; the 'Basic Channel Information' page appears:

Figure 3-34: Basic Channel Information Page (Example, Mediant 1000)



5. Click the buttons located above the 'Basic Channel Information' screen to view additional parameters.

3.3.8.5 Replacing Modules

To replace the device's modules, you must use the Web interface in combination with physical removal and insertion of the modules. In other words, when you replace a module, you first need to 'software-remove' it, then extract it physically from the chassis and insert a new module, and then 'software-insert' it using the Web interface. The software removal and insertion is performed in the 'Home' page.

Warnings:

- Replacing a damaged module can be performed only with the same type of module and in the same module slot number. For example, a module with two digital spans in Slot 1 must be replaced with a module with two digital spans in Slot 1.
- When only one module is available, removal of the module causes the device to reset.
- Adding a module to a previously empty slot must only be performed when the power to the device is switched off (refer to Inserting Modules into Previously Empty Slots).

Note: This section is applicable only to Mediant 1000.

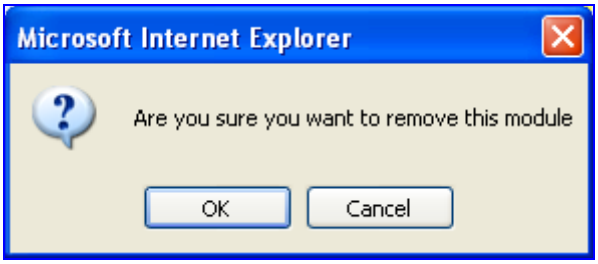
➢ To replace a module, take these 2 steps:

1. Remove the module by performing the following:
   a. In the 'Home' page, click the title of the module that you want to replace; the **Remove Module** button appears:

Figure 3-35: Remove Module Button Appears after Clicking Module Name

   b. Click the **Remove Module** button; a message box appears requesting you to confirm module removal:

Figure 3-36: Module Removal Confirmation Message Box



c. Click **OK** to confirm removal; after a few seconds, the module is software-removed, the module status icon turns to grey, and the name of the module is suffixed with the word 'Reserved':

Figure 3-37: Removed Module

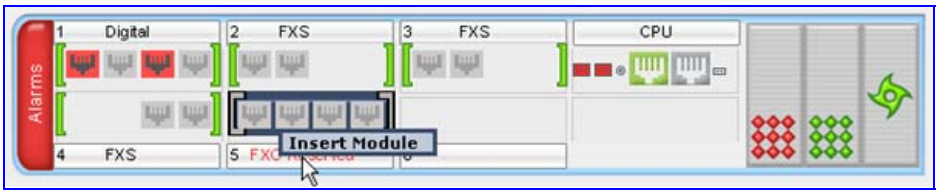
d. Physically remove the module (refer to the device's *Fast Track Installation Guide*).

2. Insert the replaced module, by performing the following:

a. Physically insert the replaced module (refer to the device's *Fast Track Installation Guide*) into the same slot in which the previous module resided.

b. In the 'Home' page, click the title of the module ('<module type> Reserved') that you want to replace; the **Insert Module** button appears:

Figure 3-38: Insert Module Button after Clicking Module's Name



c. Click the **Insert Module** button; a message appears informing you that this may take a few seconds. When the message disappears, the module is inserted, which is indicated by the disappearance of the word 'Reserved' from the module's name.

3.3.9 Logging Off the Web Interface

You can log off the Web interface and re-access it with a different user account. For detailed information on the Web User Accounts, refer to User Accounts.

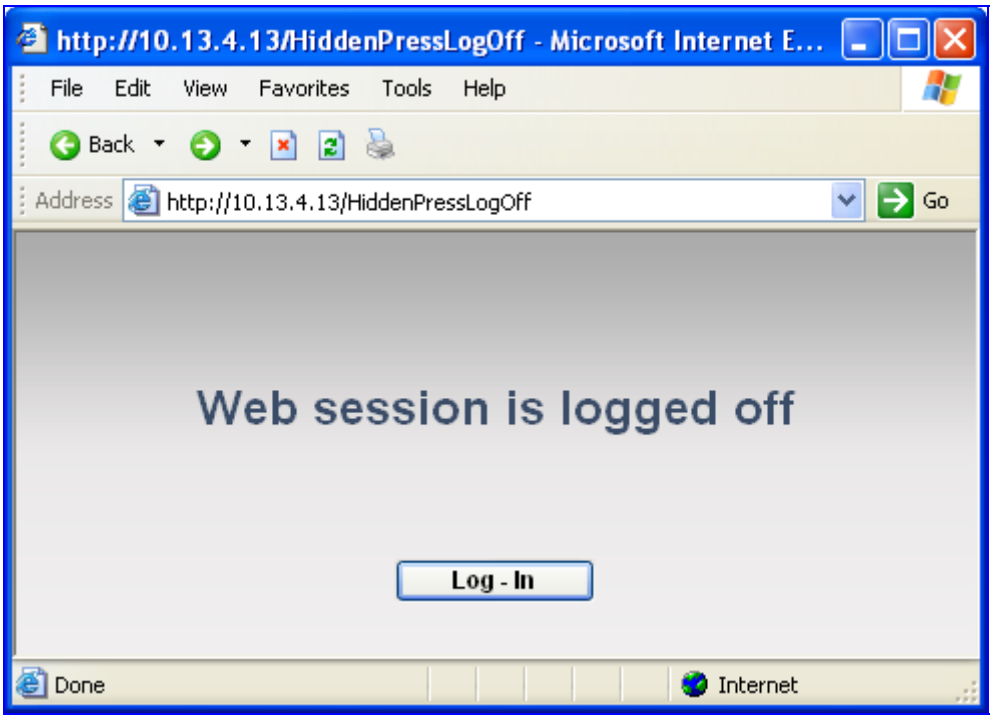
➢ To log off the Web interface, take these 2 steps:

1. On the toolbar, click the **Log Off** button; the 'Log Off' confirmation message box appears:

Figure 3-39: Log Off Confirmation Box

2. Click **OK**; the Web session is logged off and the **Log In** button appears.

Figure 3-40: Web Session Logged Off



To log in again, simply click the **Log In** button, and then in the 'Enter Network Password' dialog box, enter your user name and password (refer to "Accessing the Web Interface" on page 26).

3.4 Configuration Tab

The **Configuration** tab on the Navigation bar displays all menus related to device configuration. These menus appear in the Navigation tree and include the following:

- Network Settings (refer to "Network Settings" on page 64)
- Media Settings (refer to "Media Settings" on page 80)
- PSTN Settings (refer to PSTN Settings on page 96)
- Security Settings (refer to "Security Settings" on page 111)
- Protocol Configuration (refer to "Protocol Configuration" on page 133)
- Advanced Applications (refer to "Advanced Applications" on page 244)
- TDM Configuration (refer to "Configuring the TDM Bus Settings" on page 251)

➢ To access the menus of the Configuration tab, take this step:

- On the Navigation bar, click the **Configuration** tab; the Navigation tree displays the configuration menus pertaining to the **Configuration** tab.

3.4.1 Network Settings

The **Network Settings** menu allows you to configure various networking parameters. This menu contains the following page items:

- IP Settings (refer to "Configuring the IP Settings" on page 64)
- Application Settings (refer to "Configuring the Application Settings" on page 71)
- IP Routing Table (refer to "Configuring the IP Routing Table" on page 75)
- QoS Settings (refer to "Configuring the QoS Settings" on page 78)

3.4.1.1 Configuring the IP Settings

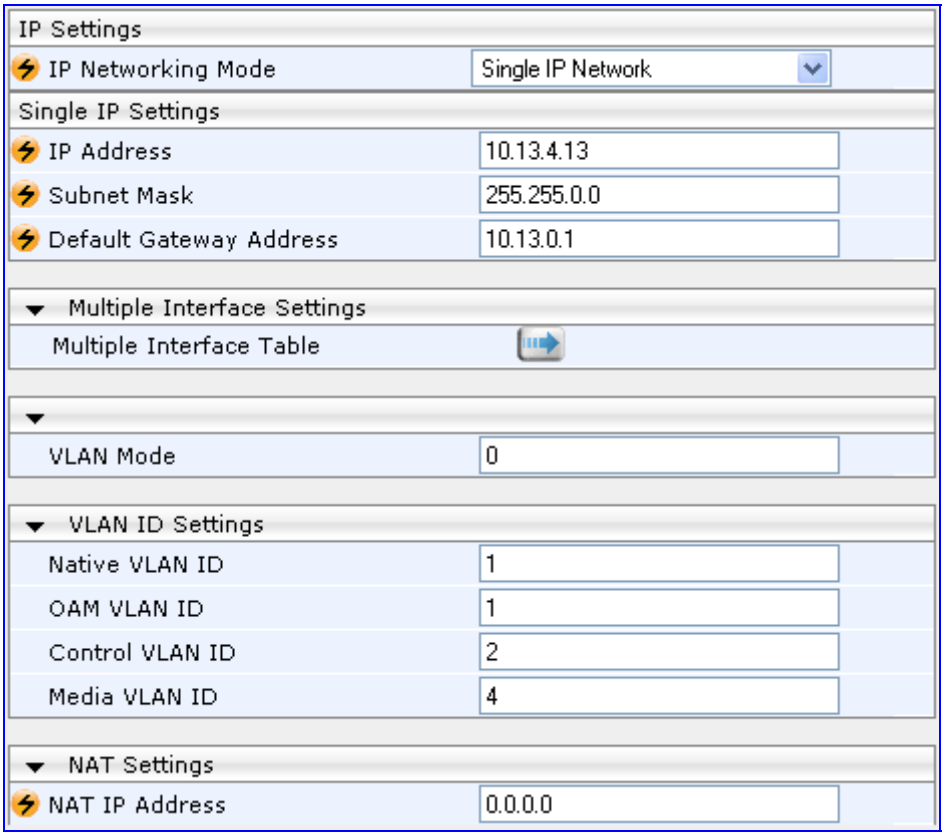
The 'IP Settings' page is used for configuring basic IP networking parameters such as the device's IP address. However, from this page you can also access the 'Multiple Interface Table' page for configuring multiple interfaces.

Note: Once you configure multiple interfaces in the 'Multiple Interface Table' page (accessed by clicking the button), when clicking the **IP Settings** page item in the Navigation tree, the 'Multiple Interface Table' page is accessed (instead of the 'IP Settings' page).

➢ To configure the IP settings parameters, take these 4 steps:

1. Open the 'IP Settings' page (**Configuration** tab > **Network Settings** menu > **IP Settings** page item).

Figure 3-41: IP Settings Page



2. Configure the IP parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-9: Network Settings -- IP Settings Parameters

Parameter	Description
IP Settings	
IP Networking Mode **[EnableMultipleIPs]**	Determines the IP network scheme. ▪ **[0]** Single IP Network = Single IP network (default). ▪ **[1]** Multiple IP Networks = Multiple IP networks (OAMP, Media, and Control). ▪ **[1]** Dual IP (Media & Control) = Multiple IP networks. ▪ **[1]** Dual IP (OAM & Control) = Multiple IP networks. ▪ **[1]** Dual IP (OAM & Medial) = Multiple IP networks. **Note:** This parameter is not relevant when using Multiple Interface tables, activated by clicking the **Multiple Interface Table** button described below (refer to "Configuring the Multiple Interface Table" on page 68").For detailed information on Multiple IPs, refer to "Multiple IPs" on page 450.
Single IP Settings	
IP Address	IP address of the device. Enter the IP address in dotted-decimal notation, for example, 10.8.201.1. **Notes:** ▪ A warning message is displayed (after clicking **Submit**) if the entered value is incorrect. ▪ After changing the IP address, you must reset the device.
Subnet Mask	Subnet mask of the device. Enter the subnet mask in dotted-decimal notation, for example, 255.255.0.0. **Notes:** ▪ A warning message is displayed (after clicking **Submit**) if the entered value is incorrect. ▪ After changing the subnet mask, you must reset the device.
Default Gateway Address	IP address of the default Gateway used by the device. Enter the IP address in dotted-decimal notation, for example, 10.8.0.1. **Notes:** ▪ A warning message is displayed (after clicking **Submit**) if the entered value is incorrect. ▪ After changing the default Gateway IP address, you must reset the device. ▪ For detailed information on multiple routers support, refer to "Multiple Routers Support" on page 449.
OAM Network Settings (Available only in Multiple IP and Dual IP modes.)	
IP Address **[LocalOAMIPAddress]**	The device's source IP address in the operations, administration, maintenance, and provisioning (OAMP) network. The default value is 0.0.0.0.
Subnet Mask **[LocalOAMSubnetMask]**	The device's subnet mask in the OAMP network. The default subnet mask is 0.0.0.0.
Default Gateway Address **[LocalOAMDefaultGW]**	N/A. Use the IP Routing table instead (refer to "Configuring the IP Routing Table" on page 75).

Parameter	Description
Control Network Settings (Available only in Multiple IP and Dual IP modes.)	
IP Address **[LocalControlIPAddress]**	The device's source IP address in the Control network. The default value is 0.0.0.0.
Subnet Mask **[LocalControlSubnetMask]**	The device's subnet mask in the Control network. The default subnet mask is 0.0.0.0.
Default Gateway Address **[LocalControlDefaultGW]**	N/A. Use the IP Routing table instead (refer to "Configuring the IP Routing Table" on page 75).
Media Network Settings (Available only in Multiple IP and Dual IP modes.)	
IP Address **[LocalMediaIPAddress]**	The device's source IP address in the Media network. The default value is 0.0.0.0.
Subnet Mask **[LocalMediaSubnetMask]**	The device's subnet mask in the Media network. The default subnet mask is 0.0.0.0.
Default Gateway Address **[LocalMediaDefaultGW]**	The device's default Gateway IP address in the Media network. The default value is 0.0.0.0.
Multiple Interface Settings	
Multiple Interface Table	Click the right-pointing arrow button to open the 'Multiple Interface Table' page. For a description of configuring multiple IP interfaces, refer to "Configuring the Multiple Interface Table" on page 68.
VLAN (For detailed information on the device's VLAN implementation, refer to "VLANS and Multiple IPs" on page 450.)	
VLAN Mode **[VlANMode]**	Enables the VLAN functionality. ▪ **[0]** Disable (default). ▪ **[1]** Enable. ▪ **[2]** PassThrough = N/A. **Note:** This parameter cannot be changed on-the-fly and requires a device reset.
VALN ID Settings	
Native VLAN ID **[VLANNativeVlanID]**	Defines the native VLAN identifier (Port VLAN ID - PVID). The valid range is 1 to 4094. The default value is 1.
OAM VLAN ID **[VLANOamVlanID]**	Defines the OAMP VLAN identifier. The valid range is 1 to 4094. The default value is 1.
Control VLAN ID **[VLANControlVlanID]**	Defines the Control VLAN identifier. The valid range is 1 to 4094. The default value is 2.
Media VLAN ID **[VLANMediaVlanID]**	Defines the Media VLAN identifier. The valid range is 1 to 4094. The default value is 3.
NAT Settings	
NAT IP Address **[StaticNatIP]**	Global (public) IP address of the device to enable static Network Address Translation (NAT) between the device and the Internet.

3.4.1.2 Configuring the Multiple Interface Table

The 'Multiple Interface Table' page allows you to configure up to three logical network interfaces, each with its own IP address, unique VLAN ID (if enabled), interface name, and application types (i.e., Control, Media, and/or Operations, Administration, Maintenance and Provisioning - OAMP) permitted on the interface. In addition, this page provides VLAN-related parameters for enabling VLANs, and for defining the 'Native' VLAN ID (VLAN ID to which incoming, untagged packets are assigned). For assigning VLAN priorities and Differentiated Services (DiffServ) for the supported Class of Service (CoS), refer to "Configuring the QoS Settings" on page 78.

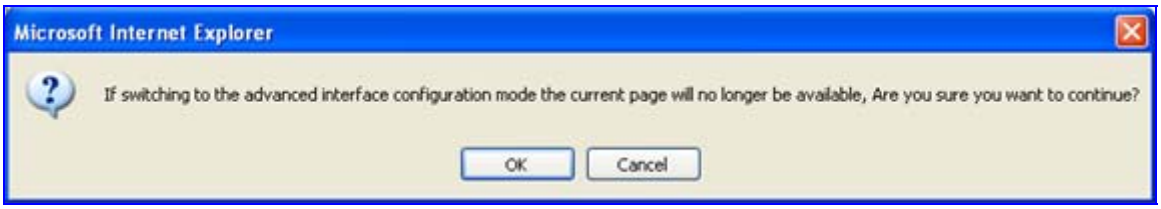
Notes:

- Once you access the 'Multiple Interface Table' page, the 'IP Settings' page is no longer available.
- You can also configure this table using the *ini* file table parameter InterfaceTable (refer to "Networking Parameters" on page 296).

➢ To configure the multiple IP interface table, take these 7 steps:

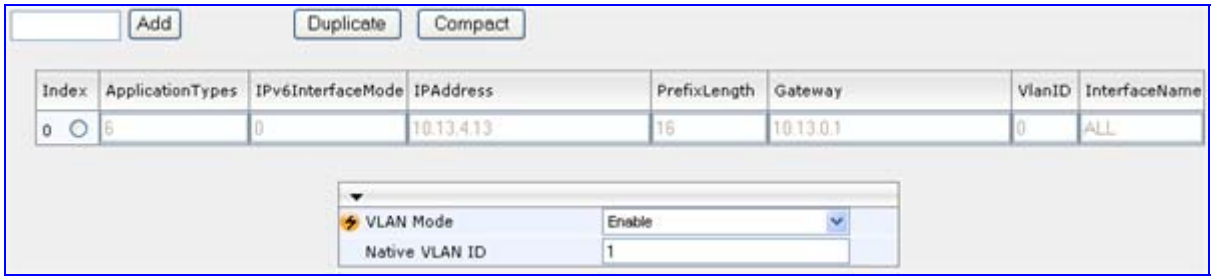
1. Open the 'IP Settings' page (refer to "Configuring the IP Settings" on page 64).
2. Under the Multiple Interface Settings group, click the right-arrow button alongside **Multiple Interface Table**; a confirmation message box appears:

Figure 3-42: Confirmation Message for Accessing the Multiple Interface Table


Microsoft Internet Explorer
If switching to the advanced interface configuration mode the current page will no longer be available, Are you sure you want to continue?
OK
Cancel

3. Click **OK** to confirm; the 'Multiple Interface Table' page appears:

Figure 3-43: Interface Table Page



4. In the 'Add' field, enter the desired index number for the new interface, and then click **Add**; the index row is added to the table.
5. Configure the interface according to the table below.
6. Click the **Apply** button; the interface is immediately applied to the device.
7. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Notes:

- When adding more than one interface to the table, ensure that you enable VLANs, using the 'VLAN Mode' (VlANMode) parameter.
- When booting using BootP/DHCP protocols (refer to the *Product Reference Manual*), an IP address is obtained from the server. This address is used as the OAMP address for this session, overriding the IP address you configured in the 'Multiple Interface Table' page. The address specified in this table takes effect only after you save the configuration to the device's flash memory. This enables the device to use a temporary IP address for initial management and configuration, while retaining the address (defined in this table) for deployment.
- For a detailed description on multiple IP interfaces and VLANs, refer to "VLANS and Multiple IPs" on page 450.
- For a description of the Web interface's table command buttons (e.g., **Duplicate** and **Delete**), refer to "Working with Tables" on page 36.

Table 3-10: Multiple Interface Table Parameters Description

Parameter	Description
Index	Index of each interface. The range is 0-3. **Note:** Each interface index must be unique.
ApplicationTypes	Types of applications that are allowed on the specific interface. ▪ **0** = (OAMP) Only Operations, Administration, Maintenance and Provisioning applications (e.g., Web, Telnet, SSH, and SNMP) are allowed on the interface. ▪ **1** = (MEDIA) Only Media (i.e., RTP streams of voice/video) is allowed on the interface. ▪ **2** = (CONTROL) Only Call Control applications (e.g., SIP) are allowed on the interface. ▪ **3** = (OAMP & MEDIA) Only OAMP and Media (RTP) applications are allowed on the interface. ▪ **4** = (OAMP & CONTROL) Only OAMP and Call Control applications are allowed on the interface. ▪ **5** = (MEDIA & CONTROL) Only Media (RTP) and Call Control applications are allowed on the interface. ▪ **6** = (ALL) All of the applications are allowed on the interface. **Notes:** ▪ Only one IPv4 interface with OAMP (0) can be configured. ▪ Only one IPv4 interface with CONTROL (2) can be configured. ▪ At least one interface with MEDIA (1) must be configured.
IPv6InterfaceMode	The IPv6 interface mode. This feature will be supported in the next applicable release. Currently, all interfaces must have the value 0 assigned to this parameter.
IPAddress	The IPv4 IP address in dotted-decimal notation. **Note:** Each interface must be assigned a unique IP address.

Parameter	Description
PrefixLength	This column lists the number of '1' bits in the subnet mask (i.e., replaces the standard dotted-decimal representation of the subnet mask for IPv4 interfaces). For example: A subnet mask of 255.0.0.0 is represented by a prefix length of 8 (i.e., 11111111 00000000 00000000 00000000), and a subnet mask of 255.255.255.252 is represented by a prefix length of 30 (i.e., 11111111 11111111 11111111 11111100). The prefix length is a Classless Inter-Domain Routing (CIDR) style presentation of a dotted-decimal subnet notation. The CIDR-style presentation is the latest method for interpretation of IP addresses. Specifically, instead of using eight-bit address blocks, it uses the variable-length subnet masking technique to allow allocation on arbitrary-length prefixes (refer to http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing for more information). The prefix length values range from 0 to 31.
Gateway	Defines the IP address of the default gateway used by the device. **Notes:** ▪ Only one default gateway can be configured for the device and it must be configured on an interface for Media traffic. All other table entries for this column must have the value 0.0.0.0. ▪ The default gateway's IP address must be in the same subnet as the interface address. ▪ For configuring additional routing rules for other interfaces, refer to "Configuring the IP Routing Table" on page 75.
VlanID	Defines the VLAN ID for each interface. When using VLANs, the VLAN ID must be unique for each interface. Incoming traffic tagged with this VLAN ID is routed to the corresponding interface, and outgoing traffic from that interface is tagged with this VLAN ID.
InterfaceName	Defines a string (up to 16 characters) to name this interface. This name is displayed in management interfaces (Web, CLI and SNMP) for better readability and has no functional use. **Note:** The interface name is a mandatory parameter and must be unique for each interface.
VLAN Mode **[VlanMode]**	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.
Native VLAN ID **[VLANNativeVlanID]**	Defines the VLAN ID to which untagged incoming traffic is assigned. Outgoing packets sent to this VLAN are sent only with a priority tag (VLAN ID = 0). When this parameter is equal to one of the VLAN IDs in the Interface Table (and VLANs are enabled), untagged incoming traffic is considered as an incoming traffic for that interface. Outgoing traffic sent from this interface is sent with the priority tag (tagged with VLAN ID = 0). When this parameter is different from any value in the 'VLAN ID' column in the Interface Table, untagged incoming traffic is discarded, and all outgoing traffic is tagged. **Note:** If this parameter is not set (i.e., default value is 1), but one of the interfaces has a VLAN ID configured to 1, this interface is still considered the 'Native' VLAN. If you do not wish to have a 'Native' VLAN ID and want to use VLAN ID 1, set this parameter to a value other than any VLAN ID in the table.

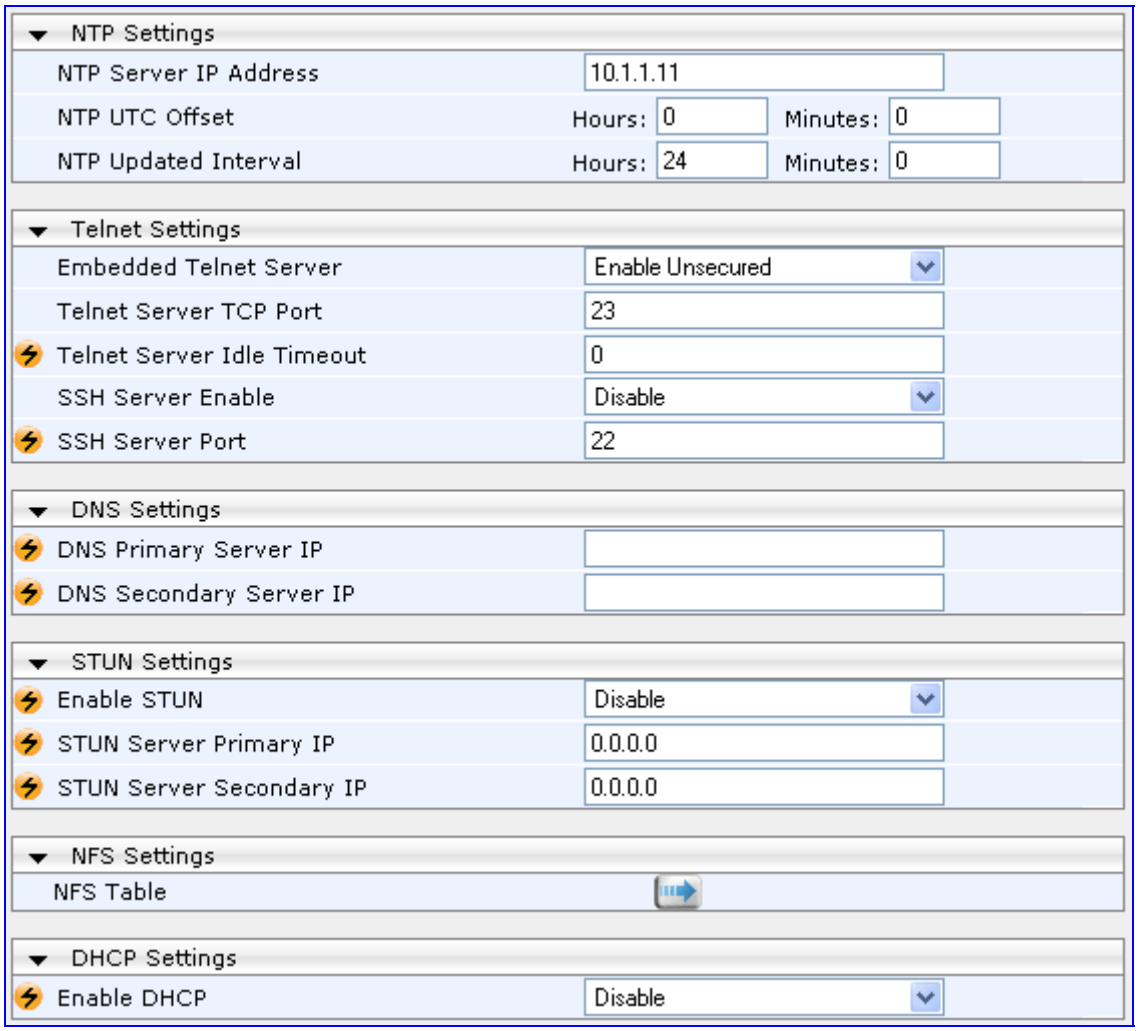
3.4.1.3 Configuring the Application Settings

The 'Application Settings' page is used for configuring various application parameters such as Telnet.

➢ To configure the Application settings parameters, take these 4 steps:

1. Open the 'Application Settings' page (**Configuration** tab > **Network Settings** menu > **Application Settings** page item).

Figure 3-44: Application Settings Page



2. Configure the Applications parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-11: Application Settings Parameters

Parameter	Description
NTP Settings (For detailed information on Network Time Protocol (NTP), refer to "Simple Network Time Protocol Support" on page 449.)	
NTP Server IP Address **[NTPServerIP]**	IP address (in dotted-decimal notation) of the NTP server. The default IP address is 0.0.0.0 (i.e., internal NTP client is disabled).
NTP UTC Offset **[NTPServerUTCOffset]**	Defines the Universal Time Coordinate (UTC) offset (in seconds) from the NTP server. The default offset is 0. The offset range is -43200 to 43200.
NTP Update Interval **[NTPUpdateInterval]**	Defines the time interval (in seconds) that the NTP client requests for a time update. The default interval is 86400 (i.e., 24 hours). The range is 0 to 214783647. **Note:** AudioCodes does not recommend setting this parameter to beyond one month (i.e., 2592000 seconds).
Telnet Settings	
Embedded Telnet Server **[TelnetServerEnable]**	Enables or disables the device's embedded Telnet server. Telnet is disabled by default for security reasons. ▪ **[0]** Disable (default) ▪ **[1]** Enable Unsecured ▪ **[2]** Enable Secured (SSL)
Telnet Server TCP Port **[TelnetServerPort]**	Defines the port number for the embedded Telnet server. The valid range is all valid port numbers. The default port is 23.
Telnet Server Idle Timeout **[TelnetServerIdleDisconnect]**	Defines the timeout (in minutes) for disconnection of an idle Telnet session. When set to zero, idle sessions are not disconnected. The valid range is any value. The default value is 0.
SSH Server Enable **[SSHServerEnable]**	Enables or disables the embedded Secure Shell (SSH) server. ▪ **[0]** Disable (default) ▪ **[1]** Enable
SSH Server Port **[SSHServerPort]**	Defines the port number for the embedded SSH server. Range is any valid port number. The default port is 23.
DNS Settings	
DNS Primary Server IP **[DNSPriServerIP]**	IP address of the primary DNS server. Enter the IP address in dotted-decimal notation, for example, 10.8.2.255. **Note:** To use Fully Qualified Domain Names (FQDN) in the 'Tel to IP Routing' table, you must define this parameter.
DNS Secondary Server IP **[DNSSecServerIP]**	IP address of the second DNS server. Enter the IP address in dotted-decimal notation, for example, 10.8.2.255.
STUN Settings	
Enable STUN **[EnableSTUN]**	Determines whether Simple Traversal of UDP through NATs (STUN) is enabled. ▪ **[0]** Disable (default) ▪ **[1]** Enable When enabled, the device functions as a STUN client and communicates with a STUN server located in the public Internet.

Parameter	Description
	STUN is used to discover whether the device is located behind a NAT and the type of NAT. In addition, it is used to determine the IP addresses and port numbers that the NAT assigns to outgoing signaling messages (using SIP) and media streams (using RTP, RTCP and T.38). STUN works with many existing NAT types and does not require any special behavior from them. For detailed information on STUN, refer to "STUN" on page 447. **Notes:** ▪ For defining the STUN server domain name, use the *ini* file parameter STUNServerDomainName (refer to "Networking Parameters" on page 296). ▪ This parameter cannot be changed on-the-fly and requires a device reset.
STUN Server Primary IP **[STUNServerPrimaryIP]**	Defines the IP address of the primary STUN server. The valid range is the legal IP addresses. The default value is 0.0.0.0.
STUN Server Secondary IP **[STUNServerSecondaryIP]**	Defines the IP address of the secondary STUN server. The valid range is the legal IP addresses. The default value is 0.0.0.0.
NFS Settings	
NFS Table	For detailed information on configuring the NFS table, refer to "Configuring the NFS Settings" on page 74.
DHCP Settings	
Enable DHCP **[DHCPEnable]**	Determines whether Dynamic Host Control Protocol (DHCP) is enabled. ▪ **[0]** Disable = Disable DHCP support on the device (default). ▪ **[1]** Enable = Enable DHCP support on the device. After the device powers up, it attempts to communicate with a BootP server. If a BootP server does not respond and if DHCP is enabled, then the device attempts to obtain its IP address and other networking parameters from the DHCP server. **Notes:** ▪ After you enable the DHCP server, perform the following procedure: 1. Click the **Submit** button, and then save the configuration (refer to "Saving Configuration" on page 265). 2. Perform a cold reset using the device's hardware reset button (soft reset via Web interface doesn't trigger the BootP/DHCP procedure and this parameter reverts to 'Disable'). ▪ Throughout the DHCP procedure the BootP/TFTP application must be deactivated, otherwise, the device receives a response from the BootP server instead of from the DHCP server. ▪ For additional information on DHCP, refer to the *Product Reference Manual*. ▪ DHCPEnable is a special 'Hidden' parameter. Once defined and saved in flash memory, its assigned value doesn't revert to its default even if the parameter doesn't appear in the *ini* file.

3.4.1.4 Configuring the NFS Settings

Network File System (NFS) enables the device to access a remote server's shared files and directories, and to handle them as if they're located locally. You can configure up to five different NFS file systems. As a file system, the NFS is independent of machine types, OSs, and network architectures. NFS is used by the device to load the *cmp*, *ini*, and auxiliary files, using the Automatic Update mechanism (refer to Automatic Update Mechanism). Note that an NFS file server can share multiple file systems. There must be a separate row for each remote file system shared by the NFS file server that needs to be accessed by the device.

➢ To add remote NFS file systems, take these 6 steps:

1. Open the 'Application Settings' page (refer to "Configuring the Application Settings" on page 71).
2. Under the NFS Settings group, click the right-arrow button alongside **NFS Table**; the 'NFS Settings' page appears.

Figure 3-45: NFS Settings Page

3. In the 'Add' field, enter the index number of the remote NFS file system, and then click **Add**; an empty entry row appears in the table.
4. Configure the NFS parameters according to the table below.
5. Click the **Apply** button; the remote NFS file system is immediately applied, which can be verified by the appearance of the 'NFS mount was successful' message in the Syslog server.
6. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Notes:

- To avoid terminating current calls, a row must not be deleted or modified while the device is currently accessing files on that remote NFS file system.
- The combination of 'HostOrIP' and 'RootPath' must be unique for each row in the table. For example, the table must include only one row with a Host / IP of 192.168.1.1 and Root Path of /audio.
- For a description of the web interface's table command buttons (e.g., **Duplicate** and **Delete**), refer to "Working with Tables" on page 36.
- You can also configure the NFS table using the *ini* file table parameter NFSServers (refer to "Networking Parameters" on page 296).

Table 3-12: Network Settings -- NFS Settings Parameters

Parameter	Description
Index	The row index of the remote file system. The valid range is 0 to 4.
HostOrIP	The domain name or IP address of the NFS server. If a domain name is provided, a DNS server must be configured.
RootPath	Path to the root of the remote file system in the format: /[path]. For example, '/audio'.
NFSVersion	NFS version used with the remote file system: 2 or 3 (default).
AuthType	Authentication method used with the remote file system. ▪ **[0]** Auth NULL. ▪ **[1]** Auth UNIX (default).
UID	User ID used in authentication when using Auth UNIX. The valid range is 0 to 65537. The default is 0.
GID	Group ID used in authentication when using Auth UNIX. The valid range is 0 to 65537. The default is 1.
VLANType	The VLAN type for accessing the remote file system. ▪ **[0]** = OAMP. ▪ **[1]** = Media (default). **Note:** This parameter applies only if VLANs are enabled or if Multiple IPs is configured (refer to "VLANS and Multiple IPs" on page 450).

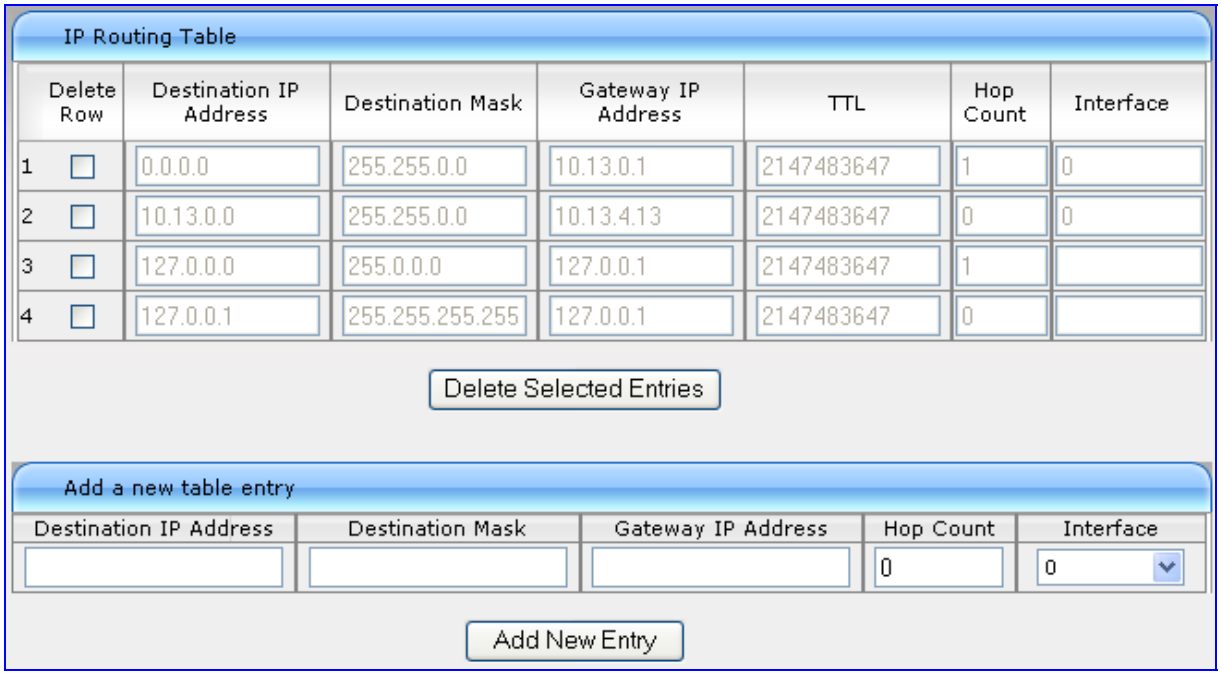
3.4.1.5 Configuring the IP Routing Table

The 'IP Routing Table' page allows you to define up to 50 static IP routing rules for the device. For example, you can define static routing rules for the OAMP and Control networks since a default gateway is supported only for the Media traffic network (refer to "Configuring the Multiple Interface Table" on page 68). Before sending an IP packet, the device searches this table for an entry that matches the requested destination host / network. If such an entry is found, the device sends the packet to the indicated router. If no explicit entry is found, the packet is sent to the default gateway (configured in the 'IP Settings' page-- refer to "Configuring the IP Settings" on page 64).

➢ To configure static IP routing, take these 3 steps:

1. Open the 'IP Routing Table' page (**Configuration** tab > **Network Settings** menu > **IP Routing Table** page item).

Figure 3-46: IP Routing Table Page



2. In the 'Add a new table entry' group, add a new static routing rule according to the parameters described in the table below.
3. Click **Add New Entry**; the new routing rule is added to the IP routing table.

To delete a routing rule from the table, select the 'Delete Row' check box that corresponds to the routing rule entry, and then click **Delete Selected Entries**.

Table 3-13: IP Routing Table Description

Column Name [ini File Parameter Name]	Description
Destination IP Address **[RoutingTableDestinationsColumn]**	Specifies the IP address of the destination host / network.
Destination Mask **[RoutingTableDestinationMasksColumn]**	Specifies the subnet mask of the destination host / network.
The address of the host / network you want to reach is determined by an AND operation that is applied to the fields 'Destination IP Address' and 'Destination Mask'. For example, to reach the network 10.8.x.x, enter 10.8.0.0 in the field 'Destination IP Address' and 255.255.0.0 in the field 'Destination Mask'. As a result of the AND operation, the value of the last two octets in the field 'Destination IP Address' is ignored. To reach a specific host, enter its IP address in the field 'Destination IP Address' and 255.255.255.255 in the field 'Destination Mask'.	

Column Name [ini File Parameter Name]	Description
Gateway IP Address **[RoutingTableGatewaysColumn]**	The IP address of the router (next hop) to which the packets are sent if their destination matches the rules in the adjacent columns. **Note:** The Gateway address must be in the same subnet on which the address is configured on the 'Multiple Interface Table' page (refer to "Configuring the Multiple Interface Table" on page 68).
TTL	A read-only field that indicates the time period for which the specific routing rule is valid. The lifetime of a static route is infinite.
Hop Count **[RoutingTableHopsCountColumn]**	The maximum number of allowed routers (hops) between the device and destination. **Note:** This parameter must be set to 1 for the routing rule to be valid. Routing entries with Hop Count equals 0 are local routes set automatically by the device.
Interface **[RoutingTableInterfacesColumn]**	Specifies the interface (network type) to which the routing rule is applied. ▪ **[0]** = OAMP (default). ▪ **[1]** = Media. ▪ **[2]** = Control. For detailed information on the network types, refer to "Configuring the Multiple Interface Table" on page 68.

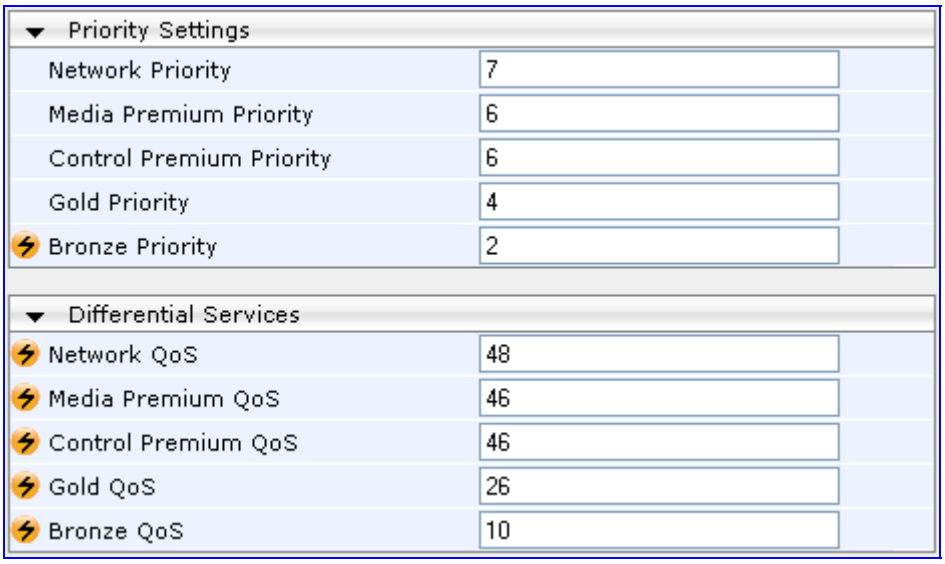
3.4.1.6 Configuring the QoS Settings

The 'QoS Settings' page is used for configuring the Quality of Service (QoS) parameters. This page allows you to assign VLAN priorities (IEEE 802.1p) and Differentiated Services (DiffServ) for the supported Class of Service (CoS).

➢ To configure QoS, take these 4 steps:

1. Open the 'QoS Settings' page (**Configuration** tab > **Network Settings** menu > **QoS Settings** page item).

Figure 3-47: QoS Settings Page



2. Configure the QoS parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-14: QoS Settings Parameters

Parameter	Description
Priority Settings	
Network Priority **[VLANNetworkServiceClassPriority]**	Defines the priority for Network Class of Service (CoS) content. The valid range is 0 to 7. The default value is 7.
Media Premium Priority **[VLANPremiumServiceClassMediaPriority]**	Defines the priority for the Premium CoS content and media traffic. The valid range is 0 to 7. The default value is 6.
Control Premium Priority **[VLANPremiumServiceClassControlPriority]**	Defines the priority for the Premium CoS content and control traffic. The valid range is 0 to 7. The default value is 6.
Gold Priority **[VLANGoldServiceClassPriority]**	Defines the priority for the Gold CoS content. The valid range is 0 to 7. The default value is 4.

Parameter	Description
Bronze Priority **[VLANBronzeServiceClassPriority]**	Defines the priority for the Bronze CoS content. The valid range is 0 to 7. The default value is 2.
Differential Services (For detailed information on IP QoS using Differentiated Services, refer to "IP QoS via Differentiated Services (DiffServ)" on page 450).	
Network QoS **[NetworkServiceClassDiffServ]**	Defines the DiffServ value for Network CoS content. The valid range is 0 to 63. The default value is 48.
Media Premium QoS **[PremiumServiceClassMediaDiffServ]**	Defines the DiffServ value for Premium Media CoS content (only if IPDiffServ is not set in the selected IP Profile). The valid range is 0 to 63. The default value is 46. **Note:** The value for the Premium Control DiffServ is determined by the following (according to priority): ▪ IPDiffServ value in the selected IP Profile. ▪ PremiumServiceClassMediaDiffServ.
Control Premium QoS **[PremiumServiceClassControlDiffServ]**	Defines the DiffServ value for Premium Control CoS content (only if ControlIPDiffserv is not set in the selected IP Profile). The valid range is 0 to 63. The default value is 40. **Note:** The value for the Premium Control DiffServ is determined by the following (according to priority): ▪ ControlPDiffserv value in the selected IP Profile. ▪ PremiumServiceClassControlDiffServ.
Gold QoS **[GoldServiceClassDiffServ]**	Defines the DiffServ value for the Gold CoS content. The valid range is 0 to 63. The default value is 26.
Bronze QoS **[BronzeServiceClassDiffServ]**	Defines the DiffServ value for the Bronze CoS content. The valid range is 0 to 63. The default value is 10.

3.4.2 Media Settings

The **Media Settings** menu allows you to configure the device's channel parameters. These parameters are applied to all the device's channels. This menu contains the following page items:

- Voice Settings (refer to "Configuring the Voice Settings" on page 81)
- Fax/Modem/CID Settings (refer to "Configuring the Fax / Modem / CID Settings" on page 84)
- RTP/RTCP Settings (refer to "Configuring the RTP / RTCP Settings" on page 88)
- General Media Settings (refer to "Configuring the General Media Settings" on page 93)
- Hook-Flash Settings (refer to "Configuring the Hook-Flash Settings" on page 94)

Notes:

- Channel parameters can be modified on-the-fly. Changes take effect from the next call.
- Some channel parameters can be configured per endpoint or call routing, using profiles (refer to "Configuring the Profile Definitions" on page 212).
- The parameter 'Fax Transport Mode' (Fax / Modem / CID Settings screen) is overridden by the parameter IsFaxUsed.

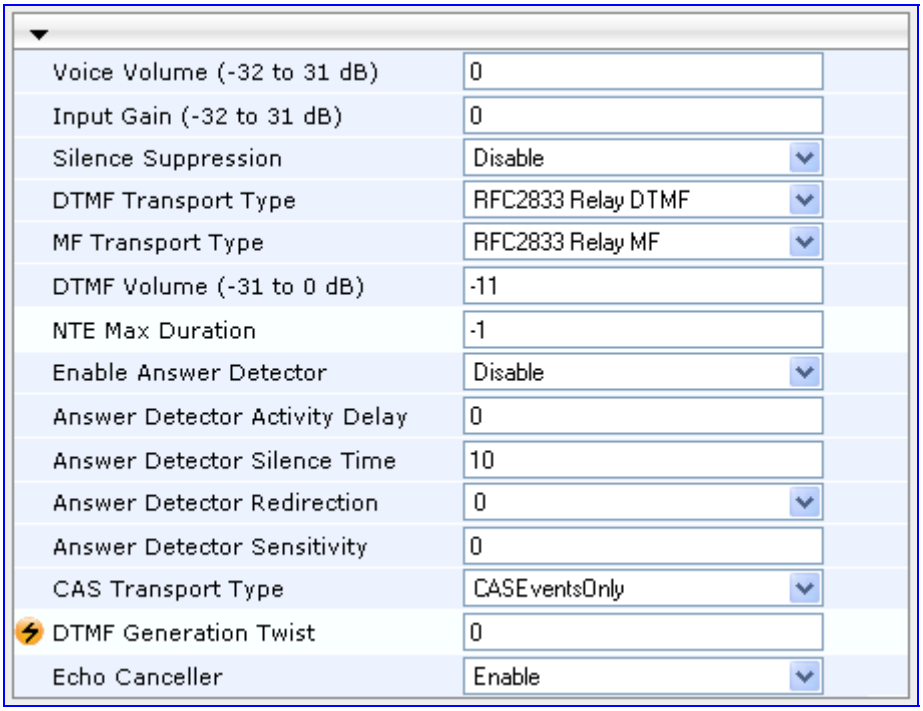
3.4.2.1 Configuring the Voice Settings

The 'Voice Settings' page is used for configuring various voice parameters such as voice volume.

➢ To configure the Voice parameters, take these 4 steps:

1. Open the 'Voice Settings' page (**Configuration** tab > **Media Settings** menu > **Voice Settings** page item).

Figure 3-48: Voice Settings Page



2. Configure the Voice parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-15: Media Settings, Voice Settings Parameters

Parameter	Description
Voice Volume **[VoiceVolume]**	Voice gain control (in decibels). This parameter sets the level for the transmitted (IP-to-PSTN/Tel) signal. The valid range is -32 to 31 dB. The default value is 0 dB.
Input Gain **[InputGain]**	Pulse-code modulation (PCM) input gain control (in decibels). This parameter sets the level for the received (Tel/PSTN-to-IP) signal. The valid range is -32 to 31 dB. The default value is 0 dB.

Parameter	Description
Silence Suppression **[EnableSilenceCompression]**	Silence Suppression is a method for conserving bandwidth on VoIP calls by not sending packets when silence is detected. ▪ **[0]** Disable = Silence Suppression is disabled (default). ▪ **[1]** Enable = Silence Suppression is enabled. ▪ **[2]** Enable without Adaptation = A single silence packet is sent during a silence period (applicable only to G.729). **Note:** If the selected coder is G.729, the following rules determine the value of the 'annexb' parameter of the fmtp attribute in the SDP: ▪ If EnableSilenceCompression is 0: 'annexb=no'. ▪ If EnableSilenceCompression is 1: 'annexb=yes'. ▪ If EnableSilenceCompression is 2 and IsCiscoSCEMode is 0: 'annexb=yes'. ▪ If EnableSilenceCompression is 2 and IsCiscoSCEMode is 1: 'annexb=no'.
Echo Canceler **[EnableEchoCanceller]**	Determines whether echo cancellation is enabled to remove echo from voice calls. ▪ **[0]** Off = Echo Canceler is disabled. ▪ **[1]** On = Echo Canceler is enabled (default). **Note:** This parameter is used to maintain backward compatibility.
DTMF Transport Type **[DTMFTransportType]**	Determines the DTMF transport type. ▪ **[0]** DTMF Mute = Erases digits from voice stream and doesn't relay to remote. ▪ **[2]** Transparent DTMF = Digits remain in voice stream. ▪ **[3]** RFC 2833 Relay DTMF = Erases digits from voice stream and relays to remote according to RFC 2833 (default). ▪ **[7]** RFC 2833 Relay Rcv Mute = DTMFs are sent according to RFC 2833 and muted when received. **Note:** This parameter is automatically updated if one of the following parameters is configured: TxDTMFOption or RxDTMFOption.
MF Transport Type **[MFTransportType]**	Not Applicable.
DTMF Volume (-31 to 0 dB) **[DTMFVolume]**	DTMF gain control value (in decibels) to the TDM or analog side. The valid range is -31 to 0 dB. The default value is -11 dB.
Enable Answer Detector **[EnableAnswerDetector]**	N/A.
Answer Detector Activity Delay **[AnswerDetectorActivityDelay]**	N/A.
Answer Detector Silence Time **[AnswerDetectorSilenceTime]**	N/A.
Answer Detector Redirection **[AnswerDetectorRedirection]**	N/A.

Parameter	Description
Answer Detector Sensitivity **[AnswerDetectorSensitivity]**	Determines the Answer Detector sensitivity. The range is 0 (most sensitive) to 2 (least sensitive). The default is 0.
CAS Transport Type **[CASTransportType]**	Controls the ABCD signaling transport type over IP. ▪ **[0]** CAS Events Only = Disable CAS relay (default). ▪ **[1]** CAS RFC2833 Relay = Enable CAS relay mode using RFC 2833. The CAS relay mode can be used with the TDM tunneling feature to enable tunneling over IP for both voice and CAS signaling bearers.
DTMF Generation Twist **[DTMFGenerationTwist]**	Defines the range (in decibels) between the high and low frequency components in the DTMF signal. Positive decibel values cause the higher frequency component to be stronger than the lower one. Negative values cause the opposite effect. For any parameter value, both components change so that their average is constant. The valid range is -10 to 10 dB. The default value is 0 dB.

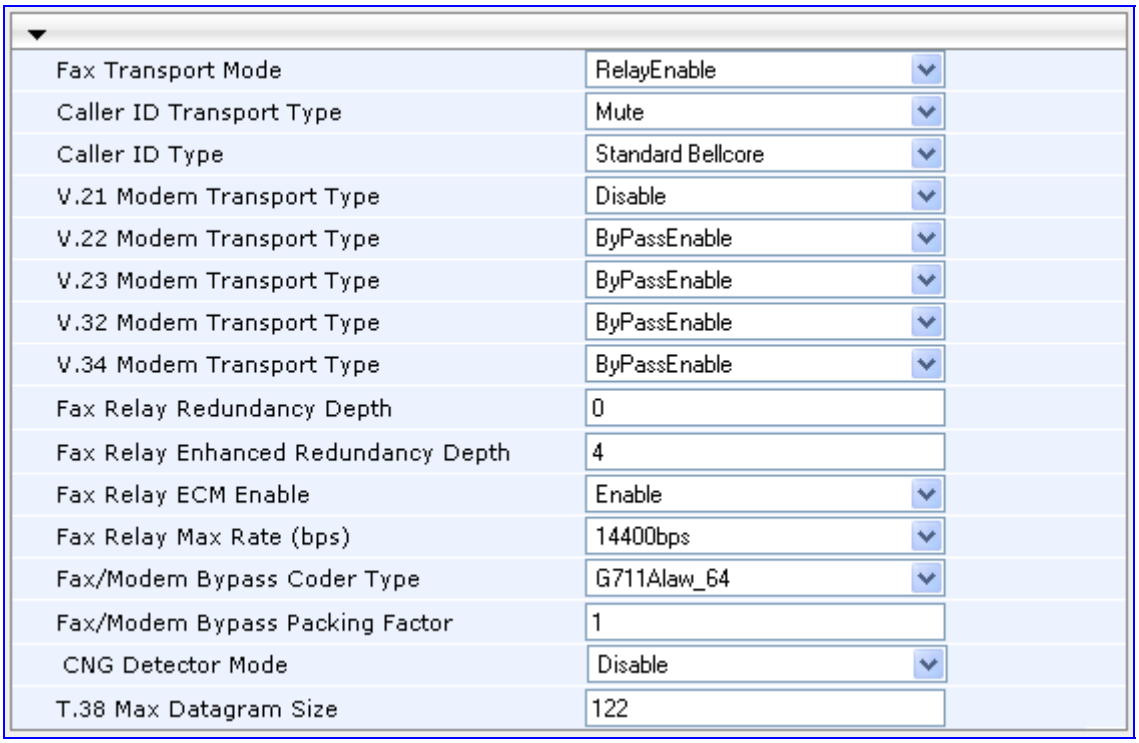
3.4.2.2 Configuring the Fax / Modem / CID Settings

The 'Fax/Modem/CID Settings' page is used for configuring fax, modem, and Caller ID (CID) parameters.

➢ **To configure the fax, modem, and CID parameters, take these 4 steps:**

1. Open the 'Fax/Modem/CID Settings' page (**Configuration** tab > **Media Settings** menu > **Fax/Modem/CID Settings** page item).

Figure 3-49: Fax/Modem/CID Settings Page



2. Configure the fax, Modem, and CID parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-16: Media Settings -- Fax/Modem/CID Parameters

Parameter	Description
Fax Transport Mode **[FaxTransportMode]**	Fax transport mode used by the device. ▪ **[0]** Disable = transparent mode. ▪ **[1]** T.38 Relay = (default). ▪ **[2]** Bypass. ▪ **[3]** Events Only. **Note:** If parameter IsFaxUsed = 1, then FaxTransportMode is always set to 1 (T.38 relay).

Parameter	Description
Caller ID Transport Type **[CallerIDTransportType]**	Determines the device's behavior for Caller ID detection. ▪ **[0]** Disable = Caller ID is not detected - DTMF digits remain in the voice stream. ▪ **[1]** Relay = Caller ID is detected - DTMF digits are erased from the voice stream. ▪ **[3]** Mute = Caller ID is detected - DTMF digits are erased from the voice stream (default).
Caller ID Type **[CallerIDType]**	Defines one of the following standards for detection (FXO) and generation (FXS) of Caller ID, and detection (FXO) and generation (FXS) of MWI (when specified) signals: ▪ **[0]** Bellcore = Caller ID and MWI (default) ▪ **[1]** ETSI = Caller ID and MWI ▪ **[2]** NTT ▪ **[4]** Britain ▪ **[16]** DTMF ETSI ▪ **[17]** Denmark = Caller ID and MWI ▪ **[18]** India ▪ **[19]** Brazil **Notes:** ▪ This parameter is applicable only to Mediant 1000. ▪ Typically, the Caller ID signals are generated/detected between the first and second rings. However, sometimes the Caller ID is detected before the first ring signal. In such a scenario, configure RingsBeforeCallerID to 0. ▪ Caller ID detection for Britain [4] is not supported on the device's FXO ports. Only FXS ports can generate the Caller ID for Britain [4]. ▪ To select the Bellcore Caller ID sub standard, use the parameter BellcoreCallerIDTypeOneSubStandard. To select the ETSI Caller ID sub-standard, use the parameter ETSICallerIDTypeOneSubStandard. ▪ To select the Bellcore MWI sub standard, use the parameter BellcoreVMWITypeOneStandard. To select the ETSI MWI sub standard, use the parameter ETSIVMWITypeOneStandard. ▪ If you define Caller ID Type as NTT [2], you need to define the NTT DID signaling form (FSK or DTMF) using NTTDIDSignallingForm.
V.21 Modem Transport Type **[V21ModemTransportType]**	V.21 Modem Transport Type used by the device. ▪ **[0]** Disable = Disable (Transparent) -- default ▪ **[1]** Enable Relay = N/A ▪ **[2]** Enable Bypass. ▪ **[3]** Events Only = Transparent with Events.
V.22 Modem Transport Type **[V22ModemTransportType]**	V.22 Modem Transport Type used by the device. ▪ **[0]** Disable = Disable (Transparent) ▪ **[1]** Enable Relay = N/A ▪ **[2]** Enable Bypass = (default) ▪ **[3]** Events Only = Transparent with Events

Parameter	Description
V.23 Modem Transport Type **[V23ModemTransportType]**	V.23 Modem Transport Type used by the device. ▪ **[0]** Disable = Disable (Transparent) ▪ **[1]** Enable Relay = N/A ▪ **[2]** Enable Bypass = (default) ▪ **[3]** Events Only = Transparent with Events
V.32 Modem Transport Type **[V32ModemTransportType]**	V.32 Modem Transport Type used by the device. ▪ **[0]** Disable = Disable (Transparent) ▪ **[1]** Enable Relay = N/A ▪ **[2]** Enable Bypass = (default) ▪ **[3]** Events Only = Transparent with Events **Note:** This option applies to V.32 and V.32bis modems.
V.34 Modem Transport Type **[V34ModemTransportType]**	V.90 / V.34 Modem Transport Type used by the device. ▪ **[0]** Disable = Disable (Transparent) ▪ **[1]** Enable Relay = N/A ▪ **[2]** Enable Bypass = (default) ▪ **[3]** Events Only = Transparent with Events
Fax Relay Redundancy Depth **[FaxRelayRedundancyDepth]**	Number of times that each fax relay payload is retransmitted to the network. ▪ **[0]** = No redundancy (default). ▪ **[1]** = One packet redundancy. ▪ **[2]** = Two packet redundancy. **Note:** This parameter is applicable only to non-V.21 packets.
Fax Relay Enhanced Redundancy Depth **[FaxRelayEnhancedRedundancyDepth]**	Number of times that control packets are retransmitted when using the T.38 standard. The valid range is 0 to 4. The default value is 2.
Fax Relay ECM Enable **[FaxRelayECMEnable]**	Determines whether the Error Correction Mode (ECM) mode is used during fax relay. ▪ **[0]** Disable = ECM mode is not used during fax relay. ▪ **[1]** Enable = ECM mode is used during fax relay (default).
Fax Relay Max Rate (bps) **[FaxRelayMaxRate]**	Maximum rate (in bps), at which fax relay messages are transmitted (outgoing calls). ▪ **[0]** 2400 = 2.4 kbps. ▪ **[1]** 4800 = 4.8 kbps. ▪ **[2]** 7200 = 7.2 kbps. ▪ **[3]** 9600 = 9.6 kbps. ▪ **[4]** 12000 = 12.0 kbps. ▪ **[5]** 14400 = 14.4 kbps (default). **Note:** The rate is negotiated between the sides (i.e., the device adapts to the capabilities of the remote side).

Parameter	Description
Fax/Modem Bypass Coder Type **[FaxModemBypassCoderType]**	Coder used by the device when performing fax/modem bypass. Usually, high-bit-rate coders such as G.711 should be used. ▪ **[0]** G.711Alaw= G.711 A-law 64 (default). ▪ **[1]** G.711Mulaw = G.711 μ-law.
Fax/Modem Bypass Packing Factor **[FaxModemBypassM]**	Number of (20 msec) coder payloads that are used to generate a fax/modem bypass packet. The valid range is 1, 2, or 3 coder payloads. The default value is 1 coder payload.
CNG Detector Mode **[CNGDetectorMode]**	Determines whether the device detects the fax Calling tone (CNG). ▪ **[0]** Disable = The originating device doesn't detect CNG; the CNG signal passes transparently to the remote side (default). ▪ **[1]** Relay = CNG is detected on the originating side. CNG packets are sent to the remote side according to T.38 (if IsFaxUsed = 1) and the fax session is started. A Re-INVITE message isn't sent and the fax session starts by the terminating device. This option is useful, for example, when the originating device is located behind a firewall that blocks incoming T.38 packets on ports that have not yet received T.38 packets from the internal network (i.e., originating device). ▪ **[2]** Events Only = CNG is detected on the originating side and a fax session is started by the originating side using the Re-INVITE message. Usually, T.38 fax session starts when the 'preamble' signal is detected by the answering side. Some SIP devices don't support the detection of this fax signal on the answering side and thus, in these cases it is possible to configure the device to start the T.38 fax session when the CNG tone is detected by the originating side. However, this mode is not recommended.

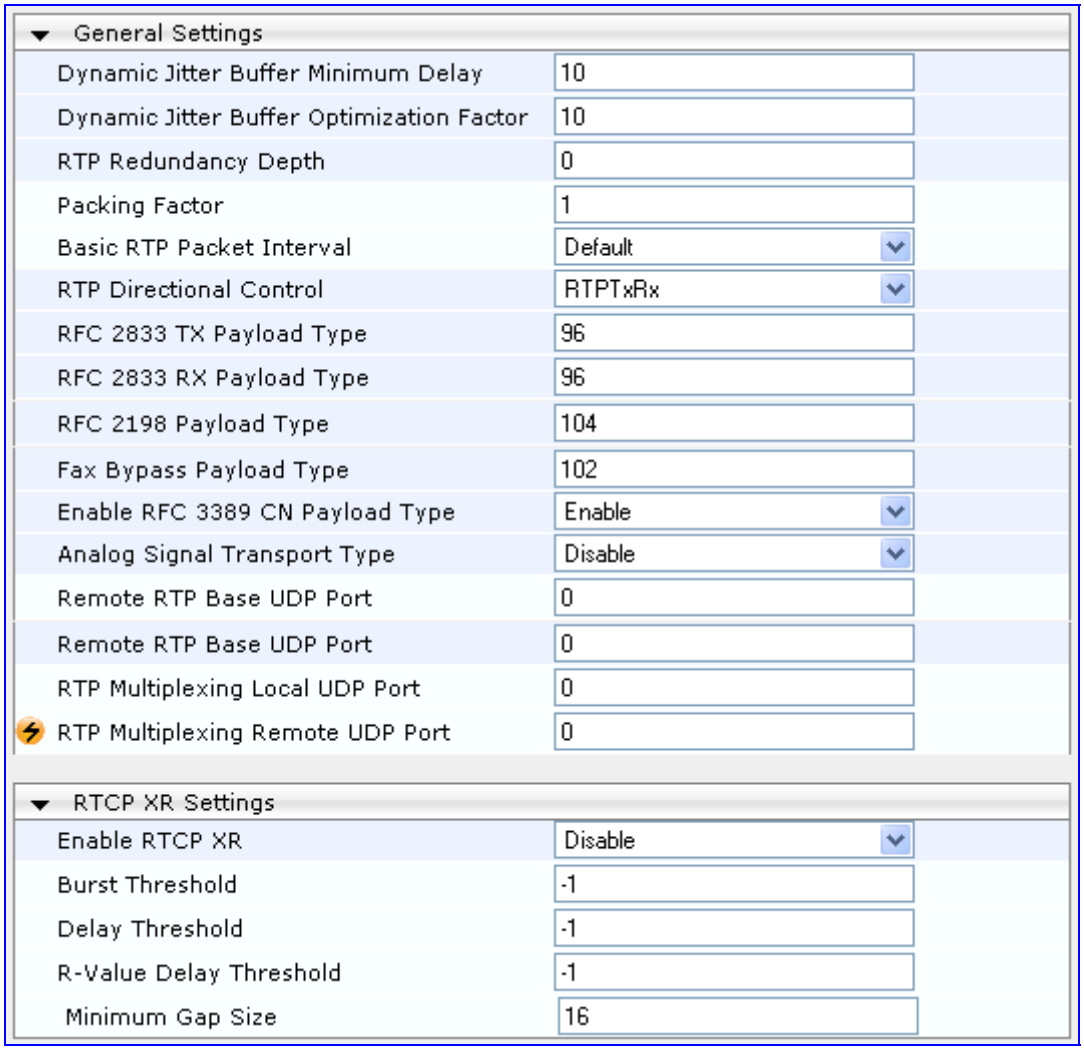
3.4.2.3 Configuring the RTP / RTCP Settings

The 'RTP/RTCP Settings' page allows you to configure the Real-Time Transport Protocol (RTP) and Real-Time Transport (RTP) Control Protocol (RTCP) parameters.

➢ **To configure the RTP / RTCP parameters, take these 4 steps:**

1. Open the 'RTP/RTCP Settings' page (**Configuration** tab > **Media Settings** menu > **RTP / RTCP Settings** page item).

Figure 3-50: RTP/RTCP Settings Page



2. Configure the RTP / RTCP parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-17: Media Settings, RTP / RTCP Parameters

Parameter	Description
Dynamic Jitter Buffer Minimum Delay **[DJBufMinDelay]**	Minimum delay (in msec) for the Dynamic Jitter Buffer. The valid range is 0 to 150. The default delay is 10. **Note:** For more information on Jitter Buffer, refer to "Dynamic Jitter Buffer Operation" on page 414.
Dynamic Jitter Buffer Optimization Factor **[DJBufOptFactor]**	Dynamic Jitter Buffer frame error / delay optimization factor. The valid range is 0 to 13. The default factor is 10. **Notes:** ▪ Set to 13 for data (fax and modem) calls. ▪ For more information on Jitter Buffer, refer to "Dynamic Jitter Buffer Operation" on page 414.
RTP Redundancy Depth **[RTPRedundancyDepth]**	Determines whether the device generates redundant packets. ▪ **[0]** 0 = Disable the generation of redundant packets (default). ▪ **[1]** 1 = Enable the generation of RFC 2198 redundancy packets.
Packing Factor **[RTPPackingFactor]**	N/A. Controlled internally by the device according to the selected coder.
Basic RTP Packet Interval **[BasicRTPPacketInterval]**	N/A. Controlled internally by the device according to the selected coder.
RTP Directional Control **[RTPDirectionControl]**	N/A. Controlled internally by the device according to the selected coder.
RFC 2833 TX Payload Type **[RFC2833TxPayloadType]**	N/A. Use the *ini* file parameter RFC2833PayloadType instead.
RFC 2833 RX Payload Type **[RFC2833RxPayloadType]**	N/A. Use the *ini* file parameter RFC2833PayloadType instead.
RFC 2198 Payload Type **[RFC2198PayloadType]**	RTP redundancy packet payload type, according to RFC 2198. The range is 96-127. The default is 104. **Note:** This parameter is applicable only if RTP Redundancy Depth = 1.
Fax Bypass Payload Type **[FaxBypassPayloadType]**	Determines the fax bypass RTP dynamic payload type. The valid range is 96 to 120. The default value is 102.
Enable RFC 3389 CN Payload Type **[EnableStandardSIDPayloadType]**	Determines whether Silence Indicator (SID) packets are sent according to RFC 3389. ▪ **[0]** Disable = G.711 SID packets are sent in a proprietary method (default). ▪ **[1]** Enable = SID (comfort noise) packets are sent with the RTP SID payload type according to RFC 3389. Applicable to G.711 and G.726 coders.

Parameter	Description
Comfort Noise Generation Negotiation **[ComfortNoiseNegotiation]**	Enables negotiation and usage of Comfort Noise (CN). ▪ **[0]** Disable = Disable (default). ▪ **[1]** Enable = Enable. The use of CN is indicated by including a payload type for CN on the media description line of the SDP. The device can use CN with a codec whose RTP timestamp clock rate is 8,000 Hz (G.711/G.726). The static payload type 13 is used. The use of CN is negotiated between sides. Therefore, if the remote side doesn't support CN, it is not used. **Note:** Silence Suppression must be enabled to generate CN.
Analog Signal Transport Type **[AnalogSignalTransportType]**	Determines the analog signal transport type. ▪ **[0]** Ignore Analog Signals = Ignore (default) ▪ **[1]** RFC2833 Analog Signal Relay = Transfer hookflash via RFC 2833 **Note:** This parameter is applicable only to Mediant 1000.
RTP Base UDP Port **[BaseUDPPort]**	Lower boundary of UDP port used for RTP, RTCP (RTP port + 1) and T.38 (RTP port + 2). The upper boundary is the Base UDP Port + 10 * (number of device's channels). The range of possible UDP ports is 6,000 to 64,000. The default base UDP port is 6000. For example: If the Base UDP Port is set to 6000 (default) then: 1) The first channel uses the following ports RTP 6000, RTCP 6001, and T.38 6002, 2) the second channel uses RTP 6010, RTCP 6011, and T.38 6012, etc. **Note:** If RTP Base UDP Port is not a factor of 10, the following message is generated: 'invalid local RTP port'. For detailed information on the default RTP/RTCP/T.38 port allocation, refer to the *Product Reference Manual*.
Remote RTP Base UDP Port **[RemoteBaseUDPPort]**	Determines the lower boundary of UDP ports used for RTP, RTCP and T.38 by a remote device. If this parameter is set to a non-zero value, ThroughPacket™ (RTP multiplexing) is enabled. The device uses this parameter (and BaseUDPPort) to identify and distribute the payloads from the received multiplexed IP packet to the relevant channels. The valid range is the range of possible UDP ports: 6,000 to 64,000. The default value is 0 (i.e., RTP multiplexing is disabled). For detailed information on RTP multiplexing, refer to RTP Multiplexing (ThroughPacket) on page 414. **Notes:** ▪ The value of this parameter on the local device must equal the value of BaseUDPPort on the remote device. ▪ To enable RTP multiplexing, the parameters L1L1ComplexTxUDPPort and L1L1ComplexRxUDPPort must be set to a non-zero value. ▪ When VLANs are implemented, RTP multiplexing is not supported.

Parameter	Description
RTP Multiplexing Local UDP Port **[L1L1ComplexTxUDPPort]**	Determines the local UDP port used for outgoing multiplexed RTP packets (applies to RTP multiplexing). The valid range is the range of possible UDP ports: 6,000 to 64,000. The default value is 0 (i.e., RTP multiplexing is disabled). This parameter cannot be changed on-the-fly and requires a device reset.
RTP Multiplexing Remote UDP Port **[L1L1ComplexRxUDPPort]**	Determines the remote UDP port to where the multiplexed RTP packets are sent, and the local UDP port used for incoming multiplexed RTP packets (applies to RTP multiplexing). The valid range is the range of possible UDP ports: 6,000 to 64,000. The default value is 0 (i.e., RTP multiplexing is disabled). This parameter cannot be changed on-the-fly and requires a device reset. **Note:** All devices that participate in the same RTP multiplexing session must use this same port.
RTCP XR Settings (**Note:** For a detailed description of RTCP XR reports, refer to the *Product Reference Manual*.)	
Enable RTCP XR **[VQMonEnable]**	Enables voice quality monitoring and RTCP Extended Reports (RTCP XR). ▪ **[0]** Disable = Disable (default) ▪ **[1]** Enable = Enables
RTCP XR Report Mode **[RTCPXRReportMode]**	Determines whether RTCP XR reports are sent to the Event State Compositor (ESC), and if so, defines the interval in which they are sent. ▪ **[0]** Disable = RTCP XR reports are not sent to the ESC (default). ▪ **[1]** End Call = RTCP XR reports are sent to the ESC at the end of each call. ▪ **[2]** End Call & Periodic = RTCP XR reports are sent to the ESC at the end of each call and periodically according to the parameter RTCPInterval.
RTCP XR Packet Interval **[RTCPInterval]**	Defines the time interval (in msec) between adjacent RTCP reports. The interval range is 0 to 65,535. The default interval is 5,000.
Disable RTCP XR Interval Randomization **[DisableRTCPRandomize]**	Controls whether RTCP report intervals are randomized or whether each report interval accords exactly to the parameter RTCPInterval. ▪ **[0]** Disable = Randomize (default) ▪ **[1]** Enable = No Randomize
RTCP XR Collection Server **[RTCPXREscIP]**	IP address of the Event State Compositor (ESC). The device sends RTCP XR reports to this server, using PUBLISH messages. The address can be configured as a numerical IP address or as a domain name.

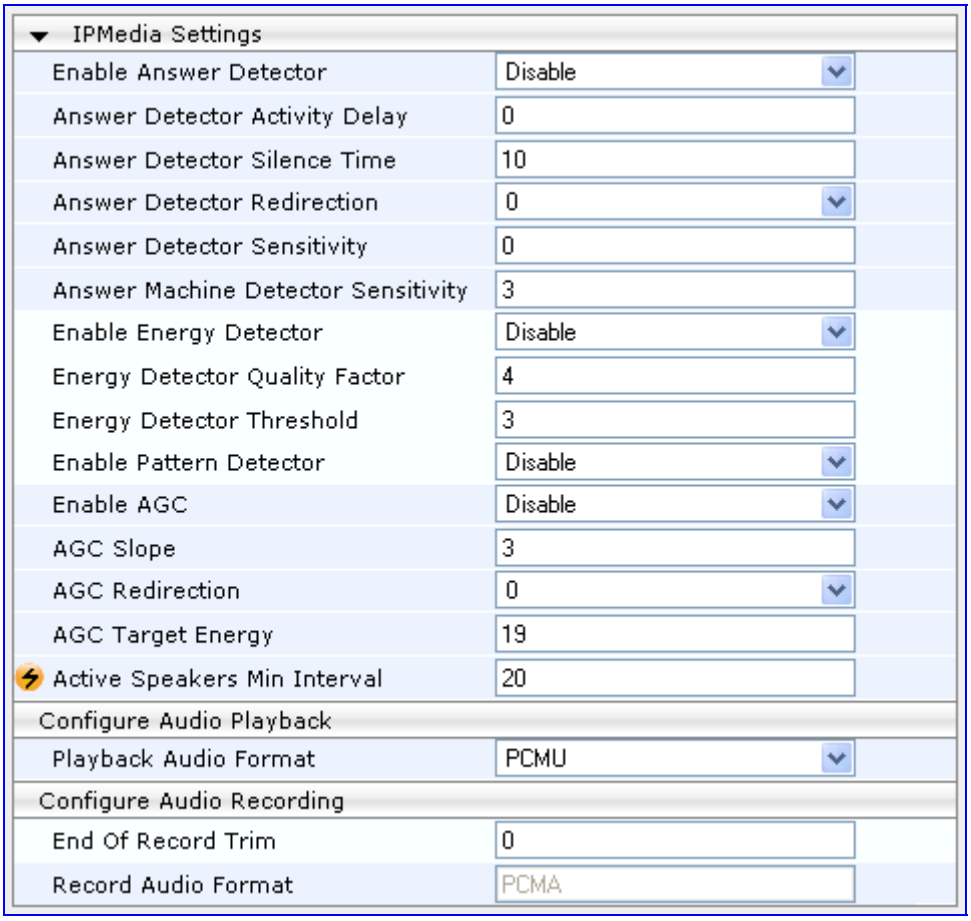
3.4.2.4 Configuring the IPmedia Settings

The 'IPMedia Settings' page allows you to configure the IP media parameters.

➢ To configure the IP media parameters, take these 4 steps:

1. Open the 'IPMedia Settings' page (**Configuration** tab > **Media Settings** menu > **IPmedia Settings** page item).

Figure 3-51: IPMedia Settings Page



2. Configure the IP media parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-18: IPMedia Parameters

Parameter	Description
Enable Answer Detector **[EnableAnswerDetector]**	N/A.
Answer Detector Activity Delay **[AnswerDetectorActivityDelay]**	N/A.
Answer Detector Silence Time **[AnswerDetectorSilenceTime]**	N/A.

Parameter	Description
Answer Detector Redirection **[AnswerDetectorRedirection]**	N/A.
Answer Detector Sensitivity **[AnswerDetectorSensitivity]**	Determines the Answer Detector sensitivity. The range is 0 (most sensitive) to 2 (least sensitive). The default is 0.
Enable Energy Detector **[EnableEnergyDetector]**	N/A
Energy Detector Quality Factor **[EnergyDetectorQualityFactor]**	N/A
Energy Detector Threshold **[EnergyDetectorThreshold]**	N/A
Enable Pattern Detector **[EnablePatternDetector]**	Enables or disables the activation of the Pattern Detector (PD). Valid options include: ▪ **[0]** Disable = Disable (default) ▪ **[1]** Enable = Enable

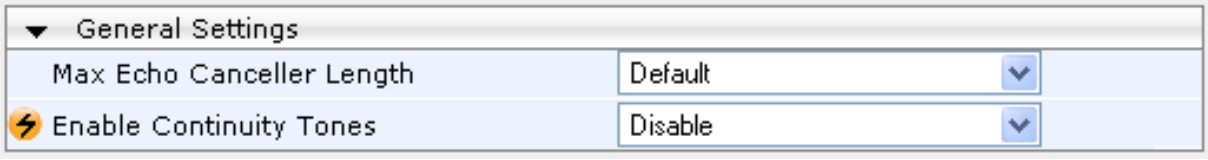
3.4.2.5 Configuring the General Media Settings

The 'General Media Settings' page allows you to configure various media parameters.

➢ To configure general media parameters, take these 4 steps:

1. Open the 'General Media Settings' page (**Configuration** tab > **Media Settings** menu > **General Media Settings** page item).

Figure 3-52: General Media Settings Page



2. Configure the general media parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-19: Media Settings Parameters

Parameter	Description
Max Echo Canceller Length **[MaxEchoCancellerLength]**	Maximum Echo Canceler Length (in msec): ▪ **[0]** Default = based on various internal device settings to attain maximum channel capacity (default) ▪ **[11]** = 64 msec ▪ **[22]** = 128 msec **Notes:** ▪ Mediant 1000 with analog / BRI module can use a max. echo canceller length of 64 msec. ▪ Mediant 1000 with PRI can use a max. echo canceller length of 128 msec. ▪ When set to 128 msec, the number of available Mediant 1000 channels is reduced by a factor of 5/6. ▪ When the Mediant 1000 houses an MPM module (in Slot #6) no channel reduction occurs. ▪ Reset the device after modifying this parameter. ▪ It isn't necessary to configure the parameter EchoCancellerLength as it automatically acquires its value from this parameter.
Enable Continuity Tones	N/A.

3.4.2.6 Configuring the Hook-Flash Settings

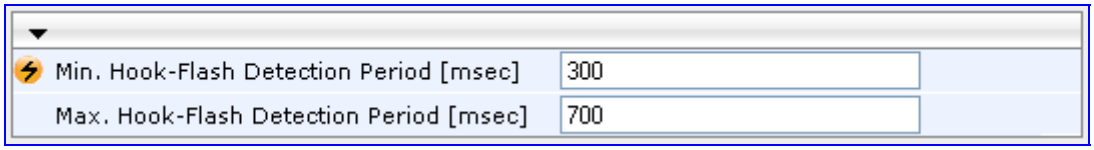
The 'Hook-Flash Settings' page allows you to configure hook-flash parameters.

Note: The 'Hook-Flash Settings' page is applicable only to Mediant 1000.

➢ **To configure the Hook-Flash parameters, take these 4 steps:**

1. Open the 'Hook-Flash Settings' page (**Configuration** tab > **Media Settings** menu > **Hook-Flash Settings** page item).

Figure 3-53: Hook-Flash Settings Screen



2. Configure the hook-flash parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-20: Hook-Flash Settings Parameters

Parameter	Description
Min. Flash-Hook Detection Period [msec] **[MinFlashHookTime]**	Defines the minimum time (in msec) for detection of a hook-flash event. The valid range is 25 to 300. The default value is 300. Detection is guaranteed for hook-flash periods of at least 60 msec (when setting the minimum time to 25). Hook-flash signals that last a shorter period of time are ignored. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ It's recommended to reduce the detection time by 50 msec from the desired value (e.g., if you want to set the value to 200 msec, then enter 150 msec (i.e., 200 minus 50).
Max. Flash-Hook Detection Period [msec] **[FlashHookPeriod]**	Defines the hook-flash period (in msec) for both analog and IP sides. For the IP side, it defines the hook-flash period that is reported to the IP. For the analog side, it defines the following: ▪ FXS interfaces: Maximum hook-flash detection period. A longer signal is considered an off-hook or on-hook event. ▪ FXS interfaces: Hook-flash generation period upon detection of a SIP INFO message containing a hook-flash signal. ▪ FXO interfaces: Hook-flash generation period. The valid range is 25 to 3,000. The default value is 700. **Note:** For FXO interfaces, a constant of 100 msec must be added to the required hook-flash period. For example, to generate a 450 msec hook-flash, set this parameter to 550.

3.4.3 PSTN Settings

The **PSTN Settings** menu allows you to configure various PSTN settings and includes the following page items:

- Trunk Settings (refer to "Configuring the Trunk Settings" on page 96)
- CAS State Machines (refer to "Configuring the CAS State Machines" on page 109)

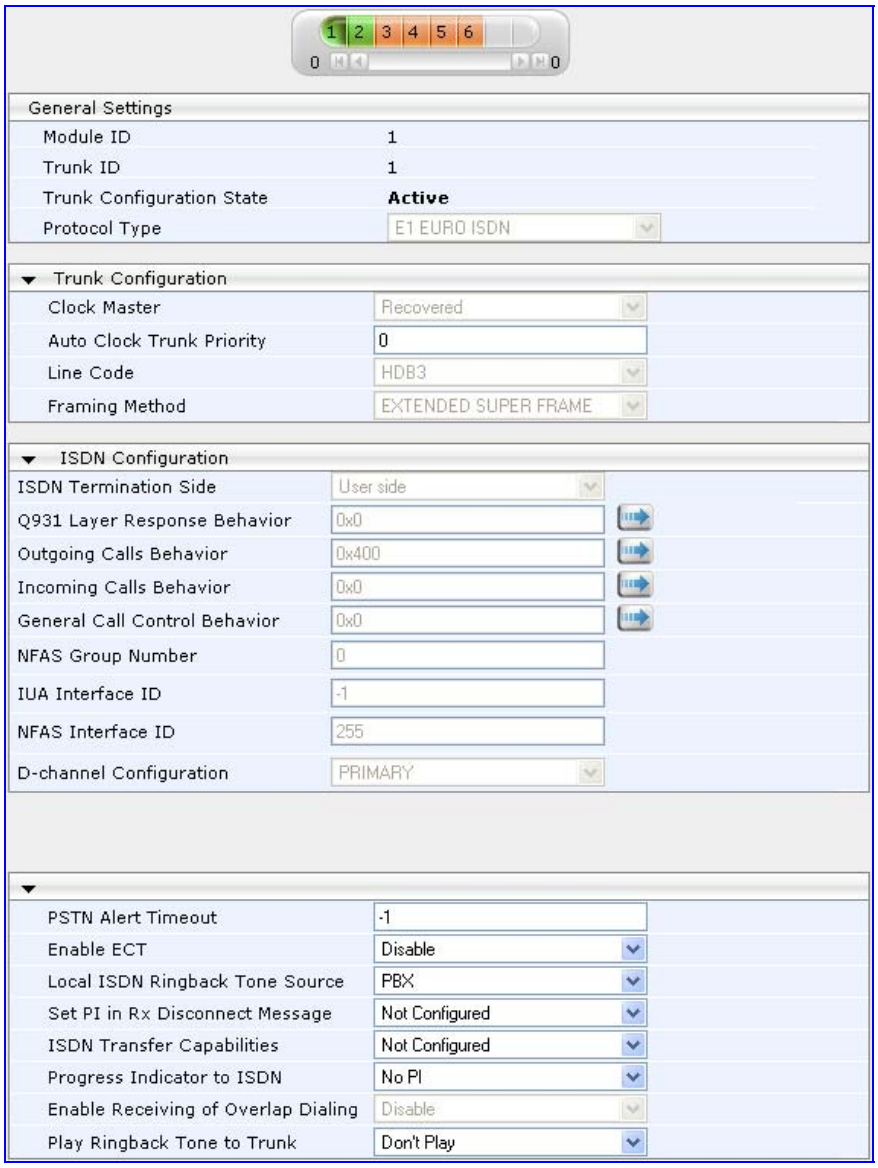
3.4.3.1 Configuring the Trunk Settings

The 'Trunk Settings' page allows you to configure the device's trunks. For configuring the trunks using the *ini* file parameters, refer to "PSTN Parameters" on page 344.

➢ To configure the Trunks, take these 7 steps:

1. Open the 'Trunk Settings' page (**Configuration** tab > **PSTN Settings** menu > **Trunk Settings** page item).

Figure 3-54: Trunk Settings Page

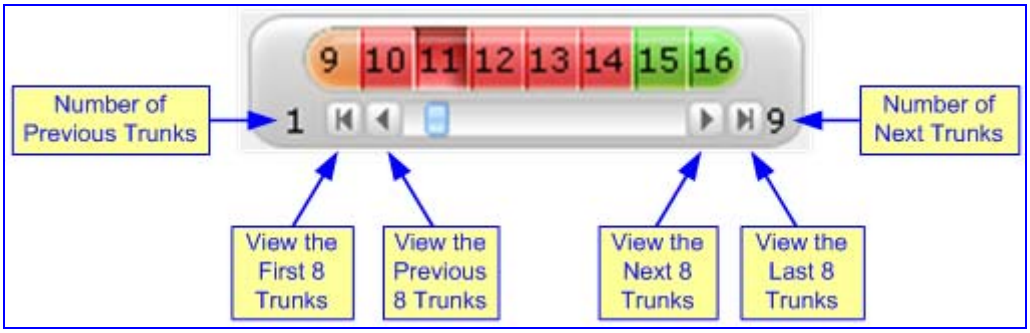

1 2 3 4 5 6
General Settings
Module ID 1
Trunk ID 1
Trunk Configuration State Active
Protocol Type E1 EURO ISDN
Trunk Configuration
Clock Master Recovered
Auto Clock Trunk Priority 0
Line Code HDB3
Framing Method EXTENDED SUPER FRAME
ISDN Configuration
ISDN Termination Side User side
Q931 Layer Response Behavior 0x0
Outgoing Calls Behavior 0x400
Incoming Calls Behavior 0x0
General Call Control Behavior 0x0
NFAS Group Number 0
IUA Interface ID -1
NFAS Interface ID 255
D-channel Configuration PRIMARY
PSTN Alert Timeout -1
Enable ECT Disable
Local ISDN Ringback Tone Source PBX
Set PI in Rx Disconnect Message Not Configured
ISDN Transfer Capabilities Not Configured
Progress Indicator to ISDN No PI
Enable Receiving of Overlap Dialing Disable
Play Ringback Tone to Trunk Don't Play

On the top of the page, a bar with Trunk number icons displays the status of each trunk, according to the following color codes:

- Grey: Disabled
- Green: Active
- Yellow: RAI alarm
- Red: LOS / LOF alarm
- Blue: AIS alarm
- Orange: D-channel alarm (ISDN only)

2. Select the trunk that you want to configure, by clicking the desired Trunk number icon. The bar initially displays the first eight trunk number icons (i.e., trunks 1 through 8). To scroll through the trunk number icons (i.e., view the next/last or previous/first group of eight trunks), refer to the figure below:

Figure 3-55: Trunk Scroll Bar


9 10 11 12 13 14 15 16
1
9
Number of Previous Trunks
Number of Next Trunks
View the First 8 Trunks
View the Previous 8 Trunks
View the Next 8 Trunks
View the Last 8 Trunks

Note: If the Trunk scroll bar displays all the available trunks, the scroll bar buttons are unavailable.

After you have selected a trunk, the following is displayed:

- The read-only 'Module ID' field displays the module number to which the trunk belongs.
- The read-only 'Trunk ID' field displays the selected trunk number.
- The read-only 'Trunk Configuration State' displays the state of the trunk (e.g., 'Active' or 'Inactive').
- The parameters displayed in the page pertain to the selected trunk only.

3. Click the **Stop Trunk** button (located at the bottom of the page) to de-activate the trunk so that you can configure currently grayed out (unavailable) parameters.(Skip this step if you want to configure parameters that are also available when the trunk is active). The stopped trunk is indicated by the following:

- The 'Trunk Configuration State' field displays 'Inactive'.
- The **Stop Trunk** button is replaced by the **Apply Trunk Settings** button. (When all trunks are stopped, the **Apply to all Trunks** button also appears.)
- All the parameters are available and can be modified.

4. Configure the desired trunk parameters, as described in the table below.

5. Click **Apply Trunk Settings** to apply the changes to the selected trunk (or click **Apply to all Trunks** to apply the changes to all trunks); the **Stop Trunk** button replaces **Apply Trunk Settings** and the 'Trunk Configuration State' displays 'Active'.
6. To save the changes to flash memory, refer to "Saving Configuration" on page 265.
7. To reset the device, refer to "Resetting the Device" on page 262.

Notes:

- If the 'Protocol Type' field displays 'NONE' (i.e., no protocol type selected) and no other trunks have been configured, after selecting a PRI protocol type, you must reset the device.
- The displayed parameters depend on the protocol selected in the 'Protocol Type' field.
- Trunks can be defined with different protocols (CAS or ISDN variants) to one another (subject to the constraints in the device's Release Notes).
- If the trunk protocol type is CAS, you can assign or modify a dial plan (in the 'Dial Plan' field) and perform this without stopping the trunk.
- If the trunk can't be stopped because it provides the device's clock (assuming the device is synchronized with the E1/T1 clock), assign a different E1/T1 trunk to provide the device's clock or enable 'TDM Bus PSTN Auto Clock' in the 'TDM Bus Settings' page (refer to "Configuring the TDM Bus Settings" on page 251).

Table 3-21: Trunk (E1/T1/J1/BRI) Configuration Parameters

ini File Field Name Web Parameter Name	Valid Range and Description
Protocol Type **[ProtocolType]**	Defines the PSTN protocol for the trunk: ▪ **[0]** = NONE ▪ **[1]** E1 EURO ISDN ▪ **[2]** T1 CAS ▪ **[3]** T1 RAW CAS ▪ **[4]** T1 TRANSPARENT ▪ **[5]** E1 TRANSPARENT 31 ▪ **[6]** E1 TRANSPARENT 30 ▪ **[7]** E1 MFCR2 ▪ **[8]** E1 CAS ▪ **[9]** E1 RAW CAS ▪ **[10]** T1 NI2 ISDN ▪ **[11]** T1 4ESS ISDN ▪ **[12]** T1 5ESS 9 ISDN ▪ **[13]** T1 5ESS 10 ISDN ▪ **[14]** T1 DMS100 ISDN ▪ **[15]** J1 TRANSPARENT ▪ **[16]** T1 NTT ISDN = Japan - Nippon Telegraph ▪ **[17]** E1 AUSTEL ISDN = Australian Telecom ▪ **[18]** T1 HKT ISDN = Hong Kong - HKT

ini File Field Name Web Parameter Name	Valid Range and Description
	▪ **[19]** E1 KOR ISDN = Korean operator ▪ **[20]** T1 HKT ISDN = Hong Kong - HKT over T1 ▪ **[21]** E1 QSIG ▪ **[23]** T1 QSIG ▪ **[31]** E1 FRENCH VN3 ISDN ▪ **[35]** T1 DMS100 Meridian ISDN ▪ **[40]** E1 NI2 ISDN ▪ **[41]** E1 CAS R15 ▪ **[50]** BRI EURO ISDN ▪ **[51]** BRI NI2 ISDN ▪ **[52]** BRI DMS100 ISDN ▪ **[53]** BRI 5ESS 10 ISDN ▪ **[54]** BRI QSIG ▪ [56] BRI NTT = BRI ISDN Japan (Nippon Telegraph) **Note:** The device simultaneously supports different variants of CAS and PRI protocols on different E1/T1 spans (no more than four simultaneous PRI variants).
Clock Master **[ClockMaster]**	Determines the Tx clock source of the E1/T1 line. ▪ **[0]** Recovered = Generate the clock according to the Rx of the E1/T1 line (default). ▪ **[1]** Generated = Generate the clock according to the internal TDM bus. **Notes:** ▪ The source of the internal TDM bus clock is determined by the parameter TDMBusClockSource. ▪ For detailed information on configuring the device's clock settings, refer to "Clock Settings" on page 459.
Auto Clock Trunk Priority **[AutoClockTrunkPriority]**	Defines the trunk priority for auto-clock fallback (per trunk parameter). ▪ 0 to 99 = priority (0 is the highest = default). ▪ 100 = the SW never performs a fallback to that trunk (usually used to mark untrusted source of clock). **Note:** Fallback is enabled when the TDMBusPSTNAutoClockEnable parameter is set to 1.
Line Code **[LineCode]**	Use to select B8ZS or AMI for T1 spans, and HDB3 or AMI for E1 spans. ▪ **[0]** B8ZS = use B8ZS line code (for T1 trunks only) default. ▪ **[1]** AMI = use AMI line code. ▪ **[2]** HDB3 = use HDB3 line code (for E1 trunks only). **Note:** This parameter is not configurable for BRI interfaces; the device automatically uses the Modified Alternate Mark Invert (MAMI) line code.

ini File Field Name Web Parameter Name	Valid Range and Description
Line Build Out Loss **[LineBuildOut.Loss]**	Defines the line build out loss for the selected T1 trunk. ▪ **[0]** 0 dB (default) ▪ **[1]** -7.5 dB ▪ **[2]** -15 dB ▪ **[3]** -22.5 dB **Note:** This parameter is not applicable for PRI E1 and BRI trunks.
Trace Level **[TraceLevel]**	Defines the trace level: ▪ **[0]** No Trace (default) ▪ **[1]** Full ISDN Trace ▪ **[2]** Layer 3 ISDN Trace ▪ **[3]** Only ISDN Q.931 Messages Trace ▪ **[4]** Layer 3 ISDN No Duplication Trace
Framing Method **[FramingMethod]**	Determines the physical framing method for the trunk. ▪ **[0]** = default according to protocol type E1 or T1. E1 default is E1 CRC4 MultiFrame Format extended G.706B (as c); T1 default is T1 Extended SuperFrame with CRC6 (as D). ▪ **[1]** = T1 SuperFrame Format (as B). ▪ **[a]** = E1 DoubleFrame Format ▪ **[b]** = E1 CRC4 MultiFrame Format ▪ **[c]** = E1 CRC4 MultiFrame Format extended G.706B ▪ **[A]** = T1 4-Frame multiframe. ▪ **[B]** = T1 12-Frame multiframe (D4). ▪ **[C]** = T1 Extended SuperFrame without CRC6 ▪ **[D]** = T1 Extended SuperFrame with CRC6 ▪ **[E]** = T1 72-Frame multiframe (SLC96) ▪ **[F]** = J1 Extended SuperFrame with CRC6 (Japan) **Note:** This parameter is not configurable for BRI interfaces; the device automatically uses the BRI framing method.
ISDN Configuration Parameters	
ISDN Termination Side **[TerminationSide]**	Selects the ISDN termination side. Applicable only to ISDN protocols. ▪ **[0]** User side = ISDN User Termination Equipment (TE) side (default) ▪ **[1]** Network side = ISDN Network Termination (NT) side **Note:** Select 'User side' when the PSTN or PBX side is configured as 'Network side' and vice versa. If you don't know the device's ISDN PRI/BRI termination side, choose 'User side'. If the D-channel alarm is indicated, choose 'Network Side'. The BRI module supports the ITU-T I.430 standard, which defines the ISDN-BRI layer 1 specification. The BRI and PRI ports are configured similarly, using this parameter. When an NT port is active, it drives a 38-V line and sends an INFO1 signal (as defined in ITU-T I.430 Table 4) on the data line to synchronize to a TE port that might be connected to it. To stop the voltage and the INFO1 signal on the line, stop the trunk using the Stop Trunk button.

ini File Field Name Web Parameter Name	Valid Range and Description
BRI Layer 2 Mode **[BriLayer2Mode]**	Determines whether Point-to-Point or Point-to-Multipoint mode for BRI ports. ▪ **[0]** Point to Point (default) ▪ **[1]** Point to Multipoint = Must be configured for Network side.
NFAS Group Number **[NFASGroupNumber_x]**	Indicates the NFAS group number (NFAS member) for the selected trunk. 'x' identifies the Trunk ID. ▪ 0 = Non NFAS trunk (default) ▪ 1 to 4 = NFAS group number Trunks that belong to the same NFAS group have the same number. With ISDN Non-Facility Associated Signaling you can use single D-channel to control multiple PRI interfaces. **Note:** This parameter is applicable only to T1 ISDN protocols.
NFAS Interface ID **[ISDNNFASInterfaceID_x]**	Defines a different Interface ID for each T1 trunk. The valid range is 0 to 100. The default interface ID equals to the trunk's ID. 'x' identifies the trunk ID. **Notes:** ▪ To set the NFAS interface ID, configure ISDNIBehavior_x to include '512' feature per T1 trunk. ▪ This parameter is not applicable to BRI interfaces.
D-channel Configuration **[DChConfig_x]**	Defines primary, backup (optional), and B-channels only. The *ini* file parameter *x* represents the Trunk ID. ▪ **[0]** PRIMARY= Primary Trunk (default) - contains a D-channel that is used for signaling. ▪ **[1]** BACKUP = Backup Trunk - contains a backup D-channel that is used if the primary D-channel fails. ▪ **[2]** NFAS = NFAS Trunk - contains only 24 B-channels, without a signaling D-channel. **Note:** This parameter is applicable only to T1 ISDN protocols.

ini File Field Name Web Parameter Name	Valid Range and Description
Enable Receiving of Overlap Dialing **[ISDNRxOverlap_x]**	Enables Rx ISDN overlap per trunk ID. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enabled. **Notes:** ▪ If enabled, the device receives ISDN called number that is sent in the 'Overlap' mode. ▪ The SETUP message to IP is sent only after the number (including the Sending Complete IE) is fully received (via SETUP and/or subsequent INFO Q.931 messages). ▪ The MaxDigits parameter can be used to limit the length of the collected number for device ISDN overlap dialing (if sending complete is not received). ▪ If a digit map pattern is defined (DigitMapping), the device collects digits until a match is found (e.g., for closed numbering schemes) or until a timer expires (e.g., for open numbering schemes). If a match is found (or the timer expires), the digit collection process is terminated even if Sending Complete wasn't received.
Local ISDN Ringback Tone Source **[LocalISDNRBSource_ID]**	Determines whether Ringback tone is played to the ISDN by the PBX / PSTN or by the device. ▪ **[0]** PBX = PBX / PSTN (default). ▪ **[1]** Gateway. This parameter is applicable to ISDN protocols. It is used simultaneously with the parameter PlayRBTone2Trunk. The *ID* in the *ini* file parameter depicts the trunk number, where 0 is the first trunk.
Progress Indicator to ISDN **[ProgressIndicator2ISDN_ID]**	Progress Indicator (PI) to ISDN. The *ID* in the *ini* file parameter depicts the trunk number, where 0 is the first trunk. ▪ **[-1]** Not Configured = The PI in ISDN messages is set according to the parameter PlayRBTone2Tel (default). ▪ **[0]** No PI = PI is not sent to ISDN. ▪ **[1]** PI = 1; **[8]** PI = 8: The PI value is sent to PSTN in Q.931/Proceeding and Alerting messages. Typically, the PSTN/PBX cuts through the audio channel without playing local Ringback tone, enabling the originating party to hear remote Call Progress Tones or network announcements.
Set PI in Rx Disconnect Message **[PIForDisconnectMsg_ID]**	Defines the device's behavior when a Disconnect message is received from the ISDN before a Connect message is received. The *ID* in the *ini* file parameter depicts the trunk number, where 0 is the first trunk. ▪ **[-1]** Not Configured = Sends a 183 SIP response according to the received progress indicator (PI) in the ISDN Disconnect message. If PI = 1 or 8, the device sends a 183 response, enabling the PSTN to play a voice announcement to the IP side. If there isn't a PI in the Disconnect message, the call is released (default). ▪ **[0]** No PI = Doesn't send a 183 response to IP. The call is released. ▪ **[1]** PI = 1; **[8]** PI = 8: Sends a 183 response to IP.

ini File Field Name Web Parameter Name	Valid Range and Description
ISDN Transfer Capabilities **[ISDNTransferCapability_ID]**	Defines the IP-to-ISDN Transfer Capability of the Bearer Capability IE in ISDN SETUP messages. The *ID* in the *ini* file parameter depicts the trunk number. ▪ **[-1]** Not Configured ▪ **[0]** Audio 3.1 = Audio (default). ▪ **[1]** Speech = Speech. ▪ **[2]** Data = Data. ▪ Audio 7 = Currently not supported. **Note:** If this parameter isn't configured or equals to '-1', Audio 3.1 capability is used.
ISDN Flexible Behavior Parameters ISDN protocol is implemented in different Switches / PBXs by different vendors. Several implementations vary a little from the specification. Therefore, to provide a flexible interface that supports these ISDN variants, the ISDN behavior parameters are used. To configure the different behavior bits in the Web interface, you can either enter the exact hexadecimal bits value in the field to the right of the relevant parameter, or directly configure each bit field by completing the following steps: 1. Click the arrow button to the right of the relevant parameter; the relevant behavior page appears. 2. Modify each bit field according to your requirements. 3. Click the **Submit** button to save your changes.	
Q.931 Layer Response Behavior **[ISDNIBehavior]**	Bit-field used to determine several behavior options that influence the behaviour of the Q.931 protocol. To select the options, click the arrow button, and then for each required option, select 1 to enable. The default is 0 (i.e., disable). ▪ **[1]** NO STATUS ON UNKNOWN IE = Q.931 Status message isn't sent if Q.931 received message contains an unknown/unrecognized IE(s). By default, the Status message is sent. **Note:** Applicable only to PRI/BRI variants in which sending of Status message is optional. ▪ **[2]** NO STATUS ON INV OP IE = Q.931 Status message isn't sent if an optional IE with invalid content is received. By default, the Status message is sent. **Note:** Applicable only to PRI/BRI variants in which sending of Status message is optional. ▪ **[4]** ACCEPT UNKNOWN FAC IE = Accepts unknown/unrecognized Facility IE. Otherwise, the Q.931 message that contains the unknown Facility IE is rejected (default). **Note:** Applicable only to PRI/BRI variants where a complete ASN1 decoding is performed on Facility IE. ▪ **[128]** SEND USER CONNECT ACK = Connect ACK message is sent in response to received Q.931 Connect. Otherwise, the Connect ACK is not sent (default). **Note:** Applicable only to Euro ISDN User side outgoing calls. ▪ **[512]** EXPLICIT INTERFACE ID = Enables to configure T1 NFAS Interface ID (refer to the parameter ISDNNFASInterfaceID_x). **Note:** Applicable to 4/5ESS, DMS, NI-2 and HKT variants. ▪ **[2048]** ALWAYS EXPLICIT = Always set the Channel Identification IE to explicit Interface ID, even if the B-channel is on the same trunk as

ini File Field Name Web Parameter Name	Valid Range and Description
	the D-channel. **Note:** Applicable to 4/5ESS, DMS and NI-2 variants. ▪ **[32768]** ACCEPT MU LAW =Mu-Law is also accepted in ETSI. ▪ **[65536]** EXPLICIT PRES SCREENING = The calling party number (octet 3a) is always present even when presentation and screening are at their default. **Note:** Applicable only to ETSI, NI-2, and 5ESS. ▪ **[131072]** STATUS INCOMPATIBLE STATE = Clears the call on receipt of Q.931 Status with incompatible state. Otherwise, no action is taken (default). ▪ **[262144]** STATUS ERROR CAUSE = Clear call on receipt of STATUS according to cause value. ▪ **[524288]** ACCEPT A LAW =A-Law is also accepted in 5ESS. ▪ **[2097152]** RESTART INDICATION =acEV_PSTN_RESTART_CONFIRM is generated on receipt of a RESTART message. ▪ **[4194304]** FORCED RESTART =On data link (re)initialization, send RESTART if there is no call. ▪ **[1073741824]** NS QSI ENCODE INTEGER = If this bit is set, INTEGER ASN.1 type is used in operator coding (compliant to new ECMA standards); otherwise, OBJECT IDENTIFIER ASN.1 type is used. **Note:** Only applicable only to QSIG. ▪ **[2147483648]** NS 5ESS NATIONAL = Use the National mode of AT&T 5ESS for B-channel maintenance. **Note:** To configure the device to support several ISDNIBehavior features, add the individual feature values. For example, to support both [512] and [2048] features, set ISDNIBehavior = 2560 (i.e., 512 + 2048).
Outgoing Calls Behavior **[ISDNOutCallsBehavior]**	This parameter determines several behaviour options that influence the behaviour of the ISDN Stack outgoing calls. To select options, click the arrow button, and then for each required option, select 1 to enable. The default is 0 (i.e., disable). ▪ **[2]** USER SENDING COMPLETE =When this bit is set, the device doesn't automatically generate the information element Sending-Complete IE in the SETUP message. If this bit is not set, the device generates it automatically in the SETUP message only. ▪ **[16]** USE MU LAW = When set, the device sends G.711-m-Law in outgoing voice calls. When disabled, the device sends G.711-A-Law in outgoing voice calls. (Applicable only to the Korean variant.) ▪ **[128]** DIAL WITH KEYPAD = When enabled, the device uses the Keypad IE to store the called number digits instead of the CALLED_NB IE. (Only applicable to the KOR variant (Korean network). Useful for Korean switches that don't accept the CALLED_NB IE.) ▪ **[256]** STORE CHAN ID IN SETUP =When this bit is set, the device forces the sending of a Channel-Id IE in an outgoing SETUP message even if it's not required by the standard (i.e., optional), and no Channel-Id has been specified in the establishment request. This is useful for improving required compatibility with switches. On BRI lines, the Channel-Id IE indicates ‘any channel’. On PRI lines, it

ini File Field Name Web Parameter Name	Valid Range and Description
	indicates an unused channel ID, preferred only. ▪ **[572]** USE A LAW = When set, the device sends G.711 A-Law in outgoing voice calls. When disabled, the device sends the default G.711-Law in outgoing voice calls. Applicable to E10 variant. ▪ **[1024]** = Numbering plan / type for T1 IP-to-Tel calling numbers are defined according to the manipulation tables or according to the RPID header (default). Otherwise, the plan / type for T1 calls are set according to the length of the calling number. ▪ **[2048]** = When this bit is set, the device accepts any IA5 character in the called_nb and calling_nb strings and sends any IA5 character in the called_nb, and is not restricted to extended digits only (i.e., 0-9,*,#). ▪ **[16384]** DLCI REVERSED OPTION = Behavior bit used in the IUA interface groups to indicate that the reversed format of the DLCI field must be used. **Note:** When using the *ini* file to configure the device to support several ISDNOutCallsBehavior features, add the individual feature values. For example, to support both [2] and [16] features, set ISDNOutCallsBehavior = 18 (i.e., 2 + 16).
Incoming Calls Behavior **[ISDNInCallsBehavior]**	This is the bit-field used to determine several behavior options that influence how the ISDN Stack INCOMING calls behave. To select the options, click the arrow button, and then for each required option, select 1 to enable. The default is 0 (i.e., disable). ▪ **[32]** DATA CONN RS = Sends a CONNECT (answer) message on NOT incoming Tel calls. ▪ **[64]** VOICE CONN RS = device sends a CONNECT (answer) message on incoming Tel calls. ▪ **[2048]** CHAN ID IN FIRST RS = Sends Channel ID in the first response to an incoming Q.931 Call Setup message. Otherwise, the Channel ID is sent only if the device requires changing the proposed Channel ID (default). ▪ **[8192]** CHAN ID IN CALL PROC = Sends Channel ID in a Q.931 Call Proceeding message. ▪ **[65536]** PROGR IND IN SETUP ACK = Includes Progress Indicator (PI=8) in Setup ACK message if an empty called number is received in an incoming SETUP message. Applicable to overlap dialing mode. The parameter also directs the device to play a dial tone (for TimeForDialTone), until the next called number digits are received. ▪ **[262144]** = NI-2 second redirect number. You can select and use (in INVITE messages) the NI-2 second redirect number if two redirect numbers are received in Q.931 Setup for incoming Tel-to-IP calls. **Note:** When using the *ini* file to configure the device to support several ISDNInCallsBehavior features, add the individual feature values. For example, to support both [2048] and [65536] features, set ISDNInCallsBehavior = 67584 (i.e., 2048 + 65536).
General Call Control Behavior **[ISDNGeneralCCBehavior]**	Bit-field used to determine several general CC behavior options. To select the options, click the arrow button, and then for each required option, select 1 to enable. The default is 0 (i.e., disable). ▪ **[2]** = data calls with interworking indication use 64 kbps B-channels (physical only).

ini File Field Name Web Parameter Name	Valid Range and Description
	▪ **[8]** REVERSE CHAN ALLOC ALGO = Channel ID allocation algorithm. ▪ **[16]** = The device clears down the call if it receives a NOTIFY message specifying 'User-Suspended'. A NOTIFY (User-Suspended) message is used by some networks (e.g., in Italy or Denmark) to indicate that the remote user has cleared the call, especially in the case of a long distance voice call. ▪ **[32]** CHAN ID 16 ALLOWED = Applies only to ETSI E1 lines (30B+D). Enables handling the differences between the newer QSIG standard (ETS 300-172) and other ETSI-based standards (ETS 300-102 and ETS 300-403) in the conversion of B-channel ID values into timeslot values: 1) In 'regular ETSI' standards, the timeslot is identical to the B-channel ID value, and the range for both is 1 to 15 and 17 to 31. The D-channel is identified as channel-id #16 and carried into the timeslot #16. 2) In newer QSIG standards, the channel-id range is 1 to 30, but the timeslot range is still 1 to 15 and 17 to 31. The D-channel is not identified as channel-id #16, but is still carried into the timeslot #16. When this bit is set, the channel ID #16 is considered as a valid B-channel ID, but timeslot values are converted to reflect the range 1 to 15 and 17 to 31. This is the new QSIG mode of operation. When this bit is not set (default), the channel_id #16 is not allowed, as for all ETSI-like standards. ▪ **[64]** USE T1 PRI = PRI interface type is forced to T1. ▪ **[128]** USE E1 PRI = PRI interface type is forced to E1. ▪ **[256]** START WITH B CHAN OOS = B-channels start in the Out-Of-Service state (OOS). ▪ **[512]** CHAN ALLOC LOWEST = CC allocates B-channels starting from the lowest available B-channel id. ▪ **[1024]** CHAN ALLOC HIGHEST = CC allocates B-channels starting from the highest available B-channel id. **Note:** When using the *ini* file to configure the device to support several ISDNGeneralCCBehavior features, add the individual feature values. For example, to support both [16] and [32] features, set ISDNGeneralCCBehavior = 48 (i.e., 16 + 32).
CAS Configuration	
CAS Table **[CASTableIndex_x]**	Defines CAS protocol for each trunk ID from a list of CAS protocols defined by the parameter CASFileName_Y. For example: CASFileName_0 = 'E_M_WinkTable.dat' CASFileName_1 = 'E_M_ImmediateTable.dat' CASTableIndex_0 = 0 CASTableIndex_1 = 0 CASTableIndex_2 = 1 CASTableIndex_3 = 1 Trunks 0 and 1 use the E&M Winkstart CAS protocol, while trunks 2 and 3 use the E&M Immediate Start CAS protocol. **Notes:** ▪ For additional CAS table *ini* file parameters (CASFileName_0, CASFileName_1, CASFileName_7, and CASTablesNum), refer to

ini File Field Name Web Parameter Name	Valid Range and Description
	"E1/T1 Configuration Parameters" on page 344. ▪ This parameter is not applicable to BRI interfaces.
Dial Plan **[CasTrunkDialPlanName]**	The Dial Plan name that is used on a specific trunk. The range is up to 11 character strings. **Note:** This parameter is not applicable to BRI interfaces.
Miscellaneous	
PSTN Alert Timeout **[TrunkPSTNAlertTimeout_ID]**	Alert Timeout (ISDN T301 timer) in seconds for outgoing calls to PSTN. This timer is used between the time that a SETUP message is sent to the Tel side (IP-to-Tel call establishment) and a CONNECT message is received. If ALERT is received, the timer is restarted. In the *ini* file parameter, *ID* depicts the trunk number, where 0 is the first trunk. The range is 1 to 600. The default is 180.
Digital Out-Of-Service Behavior **[DigitalOOSBehaviorForTrunk_ID]**	Determines the method for setting digital trunks to Out-Of-Service state per trunk. ▪ **[-1]** Not Configured = Use the settings of the DigitalOOSBehavio parameter for per device (default). ▪ **[0]** Default = Uses default behavior for each trunk (see note below). ▪ **[1]** Service = Sends ISDN In or Out of Service (only for ISDN protocols that support Service message). ▪ **[2]** D-Channel = Takes D-Channel down or up (ISDN only). ▪ **[3]** Alarm = Sends or cleans PSTN AIS Alarm (ISDN and CAS). ▪ **[4]** Block = Blocks trunk (CAS only). **Notes:** ▪ The default behavior (value 0) is as follows: - ISDN: Use Service messages on supporting variants and use Alarm on non-supporting variants. - CAS: Use Alarm. ▪ When updating this parameter value at run-time, you must stop the trunk and then restart it for the update to take effect. ▪ To determine the method for setting Out-Of-Service state for all trunks (i.e., per device), use the DigitalOOSBehavior parameter (refer to "Configuring the Digital Gateway Parameters" on page 235). ▪ The *ID* in the *ini* file parameter name represents the trunk number, where 0 is the first trunk.
Play Ringback Tone to Trunk **[PlayRBTone2Trunk_ID]**	Determines method for playing a ringback tone (RBT) to the Trunk side. In the *ini* file parameter, *ID* depicts the Trunk number, where 0 is the first trunk. ▪ **[0]** Don't Play = The device configured with ISDN / CAS protocol type, doesn't play an RBT. No PI is sent to the ISDN unless the parameter ProgressIndicator2ISDN_ID is configured differently. ▪ **[1]** Play on Local = The device configured with CAS protocol type, plays a local RBT to PSTN upon receipt of a 180 Ringing response (with or without SDP). **Note:** Receipt of a 183 response doesn't cause the device configured with CAS to play an RBT (unless SIP183Behaviour = 1). The device configured with ISDN protocol type operates according to the parameter LocalISDNRBSource:

ini File Field Name Web Parameter Name	Valid Range and Description
	1) If the device receives a 180 Ringing response (with or without SDP) and LocalISDNRBSource = 1, it plays an RBT and sends an Alert with PI = 8 (unless the parameter ProgressIndicator2ISDN_ID is configured differently). 2) If LocalISDNRBSource = 0, the device doesn't play an RBT and an Alert message (without PI) is sent to the ISDN. In this case, the PBX / PSTN should play the RBT to the originating terminal by itself. **Note:** Receipt of a 183 response doesn't cause the device with ISDN protocol type to play an RBT; the device issues a Progress message (unless SIP183Behaviour = 1). If SIP183Behaviour = 1, the 183 response is treated the same way as a 180 Ringing response. ▪ **[2]** Prefer IP = Play according to 'Early Media' (default). If a 180 response is received and the voice channel is already open (due to a previous 183 early media response or due to an SDP in the current 180 response), the device with ISDN / CAS protocol type doesn't play the RBT; PI = 8 is sent in an ISDN Alert message (unless the parameter ProgressIndicator2ISDN_ID is configured differently). If a 180 response is received, but the 'early media' voice channel is not opened, the device with CAS protocol type plays an RBT to the PSTN. The device with ISDN protocol type operates according to the parameter LocalISDNRBSource: 1) If LocalISDNRBSource = 1, the device plays an RBT and sends an ISDN Alert with PI = 8 to the ISDN (unless the parameter ProgressIndicator2ISDN_ID is configured differently). 2) If LocalISDNRBSource = 0, the device doesn't play an RBT. No PI is sent in the ISDN Alert message (unless the parameter ProgressIndicator2ISDN_ID is configured differently). In this case, the PBX / PSTN should play an RBT tone to the originating terminal by itself. **Note:** Receipt of a 183 response results in an ISDN Progress message (unless SIP183Behaviour = 1). If SIP183Behaviour = 1 (183 is handled the same way as a 180 + SDP), the device sends an Alert message with PI = 8, without playing an RBT.
B-Channel Negotiation **[BChannelNegotiationForTrunk_ID]**	Determines the ISDN B-Channel negotiation mode. ▪ **[-1]** Not Configured = use per device configuration of BChannelNegotiation parameter (default). ▪ **[0]** Preferred = Preferred. ▪ **[1]** Exclusive = Exclusive. ▪ **[2]** Any = Any. **Notes:** ▪ Applicable to ISDN protocols. ▪ The option 'Any' is only applicable if TerminationSide is set to 0 (i.e., User side). ▪ The *ID* in the *ini* file parameter name represents the trunk number, where 0 is the first trunk.

ini File Field Name Web Parameter Name	Valid Range and Description
RTP Only Mode **[RTPOnlyModeForTrunk_ID]**	Enables the device to start sending and/or receiving RTP packets to and from remote endpoints without the need to establish a Control session. The remote IP address is determined according to the 'Tel to IP Routing' table. The port is the same port as the local RTP port (configured by the parameter BaseUDPPort and the channel on which the call is received). ▪ **[-1]** Not Configured = Use the per device parameter (RTPOnlyMode) value (default). ▪ **[0]** Disable = Disabled. ▪ **[1]** Transmit & Receive = send and receive RTP packets. ▪ **[2]** Transmit Only = send RTP packets only. ▪ **[3]** Receive Only = receive RTP packets only. **Note:** The *ID* in the *ini* file parameter depicts the trunk number, where 0 is the first trunk.
Transfer Mode **[TrunkTransferMode]**	Enables the trunk Transfer Mode. Refer to TrunkTransferMode (0, 1, or 3) in "ISDN and CAS Interworking-Related Parameters" on page 348. **Note:** This parameter is only available for Protocol Type T1 CAS.
Enable TBCT **[TrunkTransferMode]**	Enables the TBCT trunk transfer mode. Refer to TrunkTransferMode (0 and 2) in "ISDN and CAS Interworking-Related Parameters" on page 348. **Note:** This parameter is only available for Protocol Type T1 N12 ISDN.
Enable RLT **[TrunkTransferMode]**	Enables the RLT trunk transfer mode. Refer to TrunkTransferMode (0 and 2) in "ISDN and CAS Interworking-Related Parameters" on page 348. **Note:** This parameter is only available for Protocol Type T1 DMS100 ISDN.
Enable Single Step Transfer **[TrunkTransferMode]**	Enables the Single Step Transfer Trunk transfer mode. Refer to TrunkTransferMode (0 and 4) in "ISDN and CAS Interworking-Related Parameters" on page 348.
Enable ECT **[TrunkTransferMode]**	Enables the ECT trunk transfer mode. Refer to TrunkTransferMode (0 and 2) in "ISDN and CAS Interworking-Related Parameters" on page 348. **Note:** This parameter is only available for Protocol Type E1 EURO ISDN.

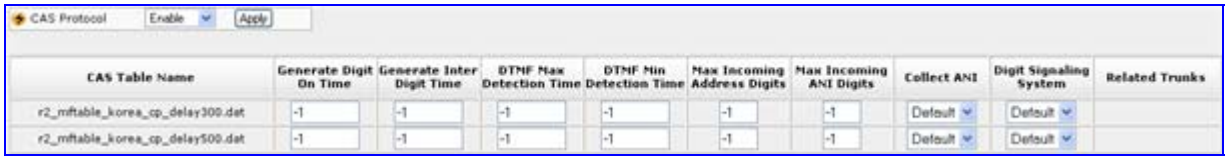
3.4.3.2 Configuring the CAS State Machines

The 'CAS State Machine' page allows you to modify various timers and other basic parameters to define the initialization of the CAS state machine without changing the state machine itself (no compilation is required). The change doesn't affect the state machine itself, but rather the configuration.

➢ To modify the CAS state machine parameters, take these 6 steps:

1. Open the 'CAS State Machine' page (**Configuration** tab > **PSTN Settings** menu > **CAS State Machines** page item).

Figure 3-56: CAS State Machine Page



2. Ensure that the trunk is inactive. The trunk number displayed in the 'Related Trunks' field must be green. If it is red (indicating that the trunk is active), click the trunk number to open the 'Trunk Settings' page (refer to "Configuring the Trunk Settings" on page 96), select the required Trunk number icon, and then click **Stop Trunk**.
3. In the 'CAS State Machine' page, modify the required parameters according to the table below.
4. Once you have completed the configuration, activate the trunk if required in the 'Trunk Settings' page, by clicking the trunk number in the 'Related Trunks' field, and in the 'Trunk Settings' page, select the required Trunk number icon, and then click **Apply Trunk Settings**.
5. Click **Submit**.
6. Reset the device (refer to "Resetting the Device" on page 262).

Notes:

- It's strongly recommended that you don't modify the default values unless you fully understand the implications of the changes and know the default values. Every change affects the configuration of the state machine parameters and the call process related to the trunk you are using with this state machine.
- You can modify CAS state machine parameters only if the following conditions are met:
  1) Trunks are inactive (stopped), i.e., the 'Related Trunks' field displays the trunk number in green.
  2) State machine is not in use or is in reset, or when it is not related to any trunk. If it is related to a trunk, you must delete the trunk or de-activate (*Stop*) the trunk.
- Field values displaying '-1' indicate CAS default values. In other words, CAS state machine values are used.
- The modification of the CAS state machine occurs at the CAS application initialization only for non-default values (-1).
- For a detailed description of the CAS Protocol table, refer to the *Product Reference Manual*.

Table 3-22: CAS State Machine Parameters Description

Parameter	Description
Generate Digit On Time **[CasStateMachineGenerateDigitOnTime]**	Generates digit on-time (in msec). The value must be a positive value. The default value is -1.
Generate Inter Digit Time **[CasStateMachineGenerateInterDigitTime]**	Generates digit off-time (in msec). The value must be a positive value. The default value is -1.
DTMF Max Detection Time **[CasStateMachineDTMFMaxOnDetectionTime]**	Detects digit maximum on time (according to DSP detection information event) in msec units. The value must be a positive value. The default value is -1.
DTMF Min Detection Time **[CasStateMachineDTMFMinOnDetectionTime]**	Detects digit minimum on time (according to DSP detection information event) in msec units. The digit time length must be longer than this value to receive a detection. Any number may be used, but the value must be less then CasStateMachineDTMFMaxOnDetectionTime. The value must be a positive value. The default value is -1.
MAX Incoming Address Digits **[CasStateMachineMaxNumOfIncomingAddressDigits]**	Defines the limitation for the maximum address digits that need to be collected. After reaching this number of digits, the collection of address digits is stopped. The value must be an integer. The default value is -1.
MAX Incoming ANI Digits **[CasStateMachineMaxNumOfIncomingANIDigits]**	Defines the limitation for the maximum ANI digits that need to be collected. After reaching this number of digits, the collection of ANI digits is stopped. The value must be an integer. The default value is -1.
Collet ANI **[CasStateMachineCollectANI]**	In some cases, when the state machine handles the ANI collection (not related to MFCR2), you can control the state machine to collect ANI or discard ANI. ▪ **[0]** No = Don't collect ANI. ▪ **[1]** Yes = Collect ANI. ▪ **[-1]** Default = Default value.
Digit Signaling System **[CasStateMachineDigitSignalingSystem]**	Defines which Signaling System to use in both directions (detection\generation). ▪ **[0]** DTMF = Uses DTMF signaling. ▪ **[1]** MF = Uses MF signaling (default). ▪ **[-1]** Default = Default value.

3.4.4 Security Settings

The **Security Settings** menu allows you to configure various security settings. This menu contains the following page items:

- Web User Accounts (refer to "Configuring the Web User Accounts" on page 112)
- Web & Telnet Access List (refer to "Configuring the Web and Telnet Access List" on page 114)
- Firewall Settings (refer to "Configuring the Firewall Settings" on page 115)

- Certificates (refer to "Configuring the Certificates" on page 118)
- General Security Settings (refer to "Configuring the General Security Settings" on page 123)
- IPSec Table (refer to "Configuring the IPSec Table" on page 126)
- IKE Table (refer to "Configuring the IKE Table" on page 130)

3.4.4.1 Configuring the Web User Accounts

To prevent unauthorized access to the Web interface, two Web user accounts are available (primary and secondary) with assigned user name and password. When you login to the Web interface, you are requested to provide this user name and password. If the Web session is idle (i.e., no actions are performed) for more than five minutes, the Web session expires and you are once again requested to login with your user name and password. Up to five Web users can simultaneously open (log in to) a session on the device's Web interface.

Each Web user account is composed of three attributes:

- **User name and password:** enables access (login) to the Web interface.
- **Access level:** determines the extent of the access (i.e., availability of pages and read / write privileges).

You can change the user name, password, and access level of each Web user account. The available access levels and their corresponding privileges are listed in the table below:

Table 3-23: Web User Accounts Access Levels and Privileges

Access Level	Numeric Representation*	Privileges
Security Administrator	200	Read / write privileges for all pages.
Administrator	100	read / write privileges for all pages except security-related pages, which are read-only.
User Monitor	50	No access to security-related and file-loading pages; read-only access to the other pages.
No Access	0	No access to any page.
* The numeric representation of the access level is used only to define accounts in a RADIUS server (the access level ranges from 1 to 255).		

Read-only access level (i.e., 'User Monitor') is typically applied to the secondary Web user account. The distribution of the primary and secondary Web user accounts to users is according to your organization's security policy.

The default attributes for the two Web user accounts are shown in the following table:

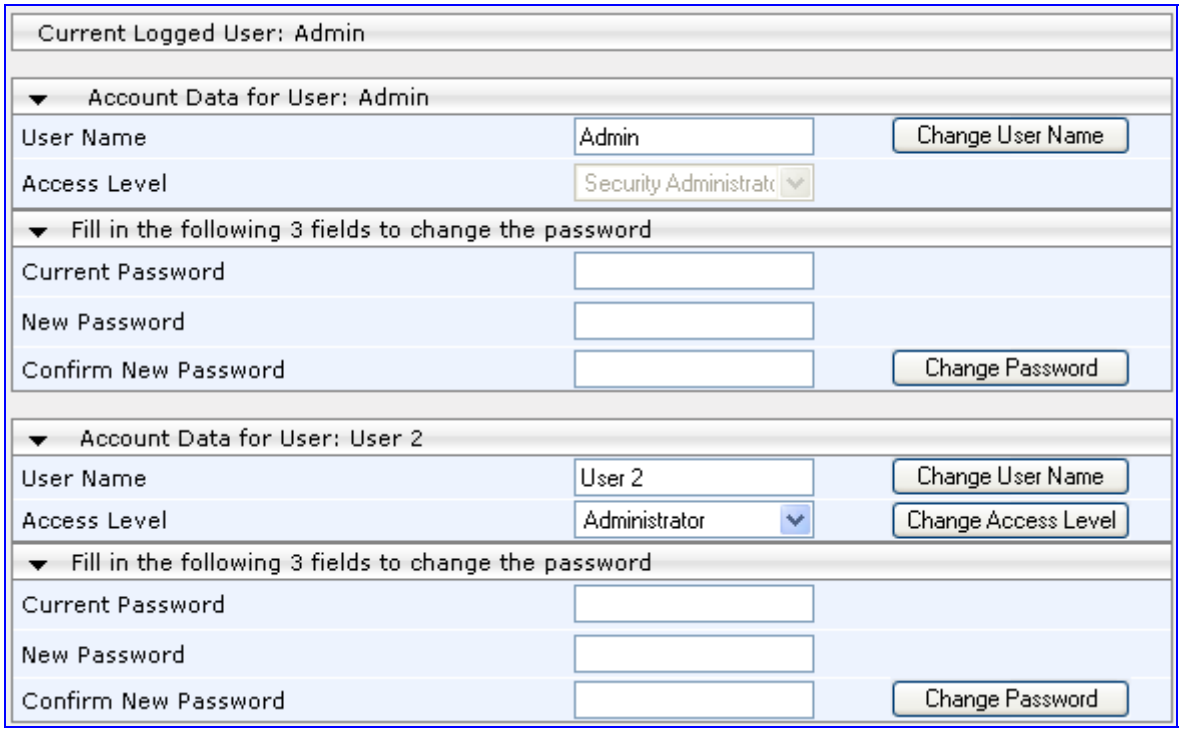
Table 3-24: Default Attributes for the Web User Accounts

Account / Attribute	User Name (Case-Sensitive)	Password (Case-Sensitive)	Access Level
Primary Account	Admin	Admin	Security Administrator
Secondary Account	User	User	User Monitor

➢ To change the Web user accounts attributes, take these 4 steps:

1. Open the 'Web User Accounts' page (**Configuration** tab > **Security Settings** menu > **Web User Accounts** page item).

Figure 3-57: Web User Accounts Page (for Users with 'Security Administrator' Privileges)



2. To change the access level of the secondary account (the access level of the primary account cannot be changed), from the 'Access Level' drop-down list, select the new access level, and then click **Change Access Level**; the new access level is applied immediately. Note that access level can only be modified by the Web user with 'Security Administrator' access level.
3. To change the user name of an account, perform the following:
   a. In the field 'User Name', enter the new user name (maximum of 19 case-sensitive characters).
   b. Click **Change User Name**; the new user name is applied immediately and the 'Enter Network Password' screen appears.
   c. In the 'Enter Network Password' screen, enter the new user name.
4. To change the password of an account, perform the following:
   a. In the field 'Current Password', enter the current password (maximum of 19 case-sensitive characters).
   b. In the fields 'New Password' and 'Confirm New Password', enter the new password.
   c. Click **Change Password**; the new password is applied immediately and the 'Enter Network Password' screen appears.
   d. In the 'Enter Network Password' screen, enter the new password.

Notes:

- For security, it's recommended that you change the default user name and password used for initially accessing the Web interface.
- The access level of the primary Web user account is 'Security Administrator', which cannot be modified; all its' other account-attributes can be modified.
- A Web user with access level 'Security Administrator' can change all attributes of all the Web user accounts. Web users with an access level other than 'Security Administrator' can only change their own password and user name.
- To access the Web interface with a different account, click the **Log off** button located on the toolbar, click any button or page item, and then re-access the Web interface with a different user name and password.
- You can set the entire Web interface to read-only (regardless of Web user account's access level), by using the *ini* file parameter DisableWebConfig (refer to "Web and Telnet Parameters" on page 311).
- To reset the two Web user accounts' user names and passwords to default, set the *ini* file parameter ResetWebPassword to 1.
- Access to the Web interface can be disabled, by setting the *ini* file parameter DisableWebTask to 1. By default, access is enabled.
- You can define additional Web user accounts using a RADIUS server (refer to the *Product Reference Manual*).
- For secured HTTP connection (HTTPS) (refer to the *Product Reference Manual*).

3.4.4.2 Configuring the Web and Telnet Access List

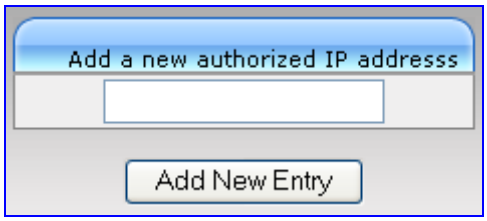
The 'Web & Telnet Access List' page is used to define up to ten IP addresses that are permitted to access the device's Web and Telnet interfaces. Access from an undefined IP address is denied. If no IP addresses are defined, this security feature is inactive and the device can be accessed from any IP address.

The Web and Telnet Access List can also be defined using the *ini* file parameter WebAccessList_x (refer to "Web and Telnet Parameters" on page 311).

➢ To add authorized IP addresses for Web and Telnet interfaces access, take these 4 steps:

1. Open the 'Web & Telnet Access List' page (**Configuration** tab > **Security Settings** menu > **Web & Telnet Access List** page item).

Figure 3-58: Web & Telnet Access List Page - Add New Entry



2. To add an authorized IP address, in the 'Add a New Authorized IP Address' field, enter the required IP address, and then click **Add New Address**; the IP address you entered is added as a new entry to the 'Web & Telnet Access List' table.

Figure 3-59: Web & Telnet Access List Table

3. To delete authorized IP addresses, select the Delete Row check boxes corresponding to the IP addresses that you want to delete, and then click **Delete Selected Addresses**; the IP addresses are removed from the table and these IP addresses can no longer access the Web and Telnet interfaces.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Notes:

- The first authorized IP address in the list must be your PC's (terminal) IP address; otherwise, access from your PC is denied.
- Only delete your PC's IP address last from the 'Web & Telnet Access List' page. If it's deleted before the last, access from your PC is denied after it's deleted.

3.4.4.3 Configuring the Firewall Settings

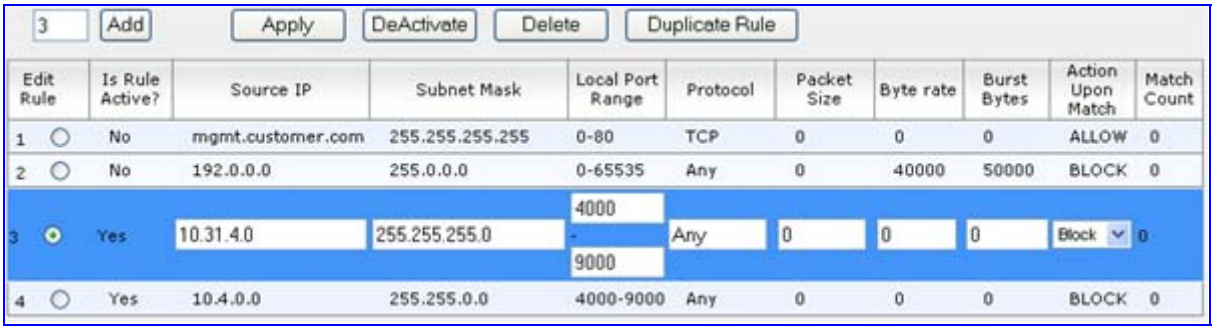
The device provides an internal firewall, allowing you (the security administrator) to define network traffic filtering rules. You can add up to 50 ordered firewall rules. For each packet received on the network interface, the table is scanned from the top down until a matching rule is found. This rule can either deny (*block*) or permit (*allow*) the packet. Once a rule in the table is located, subsequent rules further down the table are ignored. If the end of the table is reached without a match, the packet is accepted. For detailed information on the internal firewall, refer to the *Product Reference Manual*.

Note: You can also configure the firewall settings using the *ini* file table parameter AccessList (refer to "Security Parameters" on page 314).

➢ To add firewall rules, take these 5 steps:

1. Open the 'Firewall Settings' page (**Configuration** tab > **Security Settings** menu > **Firewall Settings** page item).

Figure 3-60: Firewall Settings Page



2. In the 'Add' field, enter the index of the access rule that you want to add, and then click **Add**; a new firewall rule index appears in the table.
3. Configure the firewall rule's parameters according to the table below.
4. Click one of the following buttons:
   - **Apply:** saves the new rule (without activating it).
   - **Duplicate Rule:** adds a new rule by copying a selected rule.
   - **Activate:** saves the new rule and activates it.
   - **Delete:** deletes the selected rule.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

➢ To edit a rule, take these 4 steps:

1. In the 'Edit Rule' column, select the rule that you want to edit.
2. Modify the fields as desired.
3. Click the **Apply** button to save the changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

➢ To activate a de-activated rule, take these 2 steps:

1. In the 'Edit Rule' column, select the de-activated rule that you want to activate.
2. Click the **Activate** button; the rule is activated.

➢ To de-activate an activated rule, take these 2 steps:

1. In the 'Edit Rule' column, select the activated rule that you want to de-activate..
2. Click the **DeActivate** button; the rule is de-activated.

➢ To delete a rule, take these 3 steps:

1. Select the radio button of the entry you want to activate.
2. Click the **Delete Rule** button; the rule is deleted.
3. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-25: Internal Firewall Parameters

Parameter	Description
Is Rule Active	A read-only field indicating whether the rule is active or not. **Note:** After device reset, all rules are active.
Source IP **[AccessList_Source_IP]**	IP address (or DNS name) of source network, or a specific host.
Subnet Mask **[AccessList_Net_Mask]**	IP network mask - 255.255.255.255 for a single host or the appropriate value for the source IP addresses. The IP address of the sender of the incoming packet is bitwise ANDed with this mask and then compared to the field 'Source IP'.
Local Port Range **[AccessList_Start_Port]** **[AccessList_End_Port]**	The destination UDP/TCP ports (on this device) to which packets are sent. The valid range is 0 to 65535. **Note:** When the protocol type isn't TCP or UDP, the entire range must be provided.
Protocol **[AccessList_Protocol]**	The protocol type (e.g., UDP, TCP, ICMP, ESP or 'Any'), or the IANA protocol number (in the range of 0 (Any) to 255). **Note:** This field also accepts the abbreviated strings 'SIP' and 'HTTP'. Specifying these strings implies selection of the TCP or UDP protocols, and the appropriate port numbers as defined on the device.

Parameter	Description
Packet Size **[AccessList_Packet_Size]**	Maximum allowed packet size. The valid range is 0 to 65535. **Note:** When filtering fragmented IP packets, this field relates to the overall (re-assembled) packet size, and not to the size of each fragment.
Byte Rate **[AccessList_Byte_Rate]**	Expected traffic rate (bytes per second).
Burst Bytes **[AccessList_Byte_Burst]**	Tolerance of traffic rate limit (number of bytes).
Action Upon Match **[AccessList_Allow_Type]**	Action upon match (i.e., 'Allow' or 'Block').
Match Count **[AccessList_MatchCount]**	A read-only field providing the number of packets accepted / rejected by the specific rule.

3.4.4.4 Configuring the Certificates

The 'Certificates' page is used for the following:

- Replacing the server certificate (refer to "Server Certificate Replacement" on page 118)
- Replacing the client certificates (refer to "Client Certificates" on page 120)
- Regenerating Self-Signed Certificates (refer to "Self-Signed Certificates" on page 121)
- Updating the private key (using HTTPSPkeyFileName, as described in the *Product Reference Manual*).

3.4.4.4.1 Server Certificate Replacement

The device is supplied with a working Secure Socket Layer (SSL) configuration consisting of a unique self-signed server certificate. If an organizational Public Key Infrastructure (PKI) is used, you may wish to replace this certificate with one provided by your security administrator.

➢ To replace the device's self-signed certificate, take these 8 steps:

1. Your network administrator should allocate a unique DNS name for the device (e.g., dns_name.corp.customer.com). This name is used to access the device, and should therefore be listed in the server certificate.

2. Open the 'Certificates Signing Request' page (**Configuration** tab > **Security Settings** menu > **Certificates** page item).

Figure 3-61: Certificates Signing Request Page

Certificate Signing Request
Subject Name
Generate CSR
Copy the certificate signing request and send it to your Certification Authority for signing.
Press the button "Generate self-signed" to create a self-signed certificate using the subject name provided above.
Important: this is a lengthy operation, during this time the device will be out of service.
After the operation is complete, save configuration and reset the device.
Generate self-signed
Certificate Files
Send "Server Certificate" file from your computer to the device
Browse...
Send File
Send "Trusted Root Certificate Store" file from your computer to the device
Browse...
Send file
Send "Private Key" file from your computer to the device
Browse...
Send file
Note: Replacing the private key is not recommended but if it's done, it should be over a physically-secure network link.

3. In the 'Subject Name' field, enter the DNS name, and then click **Generate CSR**. A textual certificate signing request that contains the SSL device identifier is displayed.

4. Copy this text and send it to your security provider. The security provider (also known as Certification Authority or CA) signs this request and then sends you a server certificate for the device.

5. Save the certificate to a file (e.g., cert.txt). Ensure that the file is a plain-text file containing the 'BEGIN CERTIFICATE' header, as shown in the example of a Base64-Encoded X.509 Certificate below:

```
-----BEGIN CERTIFICATE-----
MIIDkzCCAnugAwIBAgIEAgAAADANBgkqhkiG9w0BAQQFADA/MQswCQYDVQQGEwJGUj
ETMBEGA1UEChMKQ2VydGlwb3N0ZTEbMBkGA1UEAxMSQ2VydGlwb3N0ZSBTZXJ2ZXVy
MB4XDTk4MDYyNDA4MDAwMFoXDTE4MDYyNDA4MDAwMFowPzELMAkGA1UEBhMCRlIxEz
ARBgNVBAoTCkNlcnRpcG9zdGUxGzAZBgNVBAMTEkNlcnRpcG9zdGUgU2VydmV1cjCC
ASEwDQYJKoZIhvcNAQEBBQADggEOADCCAQkCggEAPqd4MziR4spWldGRx8bQrhZkon
WnNm`+Yhb7+4Q67ecf1janH7GcN/SXsfx7jJpreWULf7v7Cvpr4R7qIJcmdHIntmf7
JPM5n6cDBv17uSW63er7NkVnMFHwK1QaGFLMybFkzaeGrvFm4k3lRefiXDmuOe+FhJ
gHYezYHf44LvPRPwhSrzi9+Aq3o8pWDguJuZDIUP1F1jMa+LPwvREXfFcUW+w==
-----END CERTIFICATE-----
```

6. Set the parameter 'Secured Web Connection (HTTPS)' to 'Disable' (HTTPSOnly to 0) in "Configuring the General Security Settings" on page 123 to ensure you have a method of accessing the device in case the new certificate doesn't work. Restore the previous setting after testing the configuration.

7. In the 'Certificates Files' group, click the **Browse** button corresponding to 'Send Server Certificate...', navigate to the cert.txt file, and then click **Send File**.
8. When the loading of the certificate is completed, save the configuration (refer to "Saving Configuration" on page 265) and restart the device; the Web interface uses the provided certificate.

Notes:

- The certificate replacement process can be repeated when necessary (e.g., the new certificate expires).
- It is possible to use the IP address of the device (e.g., 10.3.3.1) instead of a qualified DNS name in the Subject Name. This is not recommended since the IP address is subject to changes and may not uniquely identify the device.
- The server certificate can also be loaded via *ini* file using the parameter HTTPSCertFileName.

3.4.4.4.2 Client Certificates

By default, Web servers using SSL provide one-way authentication. The client is certain that the information provided by the Web server is authentic. When an organizational PKI is used, two-way authentication may be desired: both client and server should be authenticated using X.509 certificates. This is achieved by installing a client certificate on the managing PC, and loading the same certificate (in base64-encoded X.509 format) to the device's Trusted Root Certificate Store. The Trusted Root Certificate file should contain both the certificate of the authorized user and the certificate of the CA.

Since X.509 certificates have an expiration date and time, the device must be configured to use NTP (refer to "Simple Network Time Protocol Support" on page 449) to obtain the current date and time. Without the correct date and time, client certificates cannot work.

➢ To enable two-way client certificates, take these 5 steps:

1. Set the parameter 'Secured Web Connection (HTTPS)' to 'Disable' (HTTPSOnly to 0) in "Configuring the General Security Settings" on page 123 to ensure you have a method of accessing the device in case the client certificate doesn't work. Restore the previous setting after testing the configuration.
2. Open the 'Certificates Signing Request' page (refer to "Server Certificate Replacement" on page 118).
3. In the 'Certificates Files' group, click the **Browse** button corresponding to 'Send "Trusted Root Certificate Store" file ...', navigate to the file, and then click **Send File**.
4. When the operation is complete, set the *ini* file parameter HTTPSRequireClientCertificates to 1.
5. Save the configuration (refer to "Saving Configuration" on page 265), and then restart the device.

When a user connects to the secured Web server:

- If the user has a client certificate from a CA that is listed in the Trusted Root Certificate file, the connection is accepted and the user is prompted for the system password.
- If both the CA certificate and the client certificate appear in the Trusted Root Certificate file, the user is not prompted for a password (thus, providing a single-sign-on experience - the authentication is performed using the X.509 digital signature).
- If the user doesn't have a client certificate from a listed CA, or doesn't have a client certificate at all, the connection is rejected.

Notes:

- The process of installing a client certificate on your PC is beyond the scope of this document. For more information, refer to your Web browser or operating system documentation, and/or consult your security administrator.
- The root certificate can also be loaded via *ini* file using the parameter HTTPSRootFileName.
- You can enable Online Certificate Status Protocol (OCSP) on the device to check whether a peer's certificate has been revoked by an OCSP server. For further information, refer to the *Product Reference Manual*.

3.4.4.4.3 Self-Signed Certificates

The device is shipped with an operational, self-signed server certificate. The subject name for this default certificate is 'ACL_nnnnnnn', where *nnnnnnn* denotes the serial number of the device. However, this subject name may not be appropriate for production and can be changed while still using self-signed certificates.

➢ To change the subject name and regenerate the self-signed certificate, take these 4 steps:

1. Before you begin, ensure the following:
   - You have a unique DNS name for the device (e.g., dns_name.corp.customer.com). This name is used to access the device and should therefore, be listed in the server certificate.
   - No traffic is running on the device. The certificate generation process is disruptive to traffic and should be executed during maintenance time.
2. Open the 'Certificates' page (refer to "Server Certificate Replacement" on page 118).
3. In the 'Subject Name' field, enter the fully-qualified DNS name (FQDN) as the certificate subject, and then click **Generate Self-signed**; after a few seconds, a message appears displaying the new subject name.
4. Save configuration (refer to "Saving Configuration" on page 265), and then restart the device for the new certificate to take effect.

Alternatively, the self-signed server certificate may be regenerated (e.g., using the subject name 'dns_name.corp.customer.com'), using the CLI command **CertificateMgmt** (CM) in the CONFiguration directory:

```
/> /CONF/CM GENERATE dns_name.corp.customer.com
```

➢ To export the current server certificate to a file using Microsoft Internet Explorer, take these 6 steps:

1. Access the device's Web interface.
2. Double-click the yellow padlock icon displayed at the bottom of the Browser's window.
3. Select the 'Details' tab, and then click **Copy to file**.
4. Click **Next**, select 'Base64-encoded X.509', and then click **Next**.
5. Select a file name, and then click **Next**.
6. Click **Finish**; the certificate is saved to the selected file name.

To export the current server certificate using CLI, type the following:

```
/> /CONF/CM GETCERT
```

The server certificate is displayed in base64-encoded PEM format.

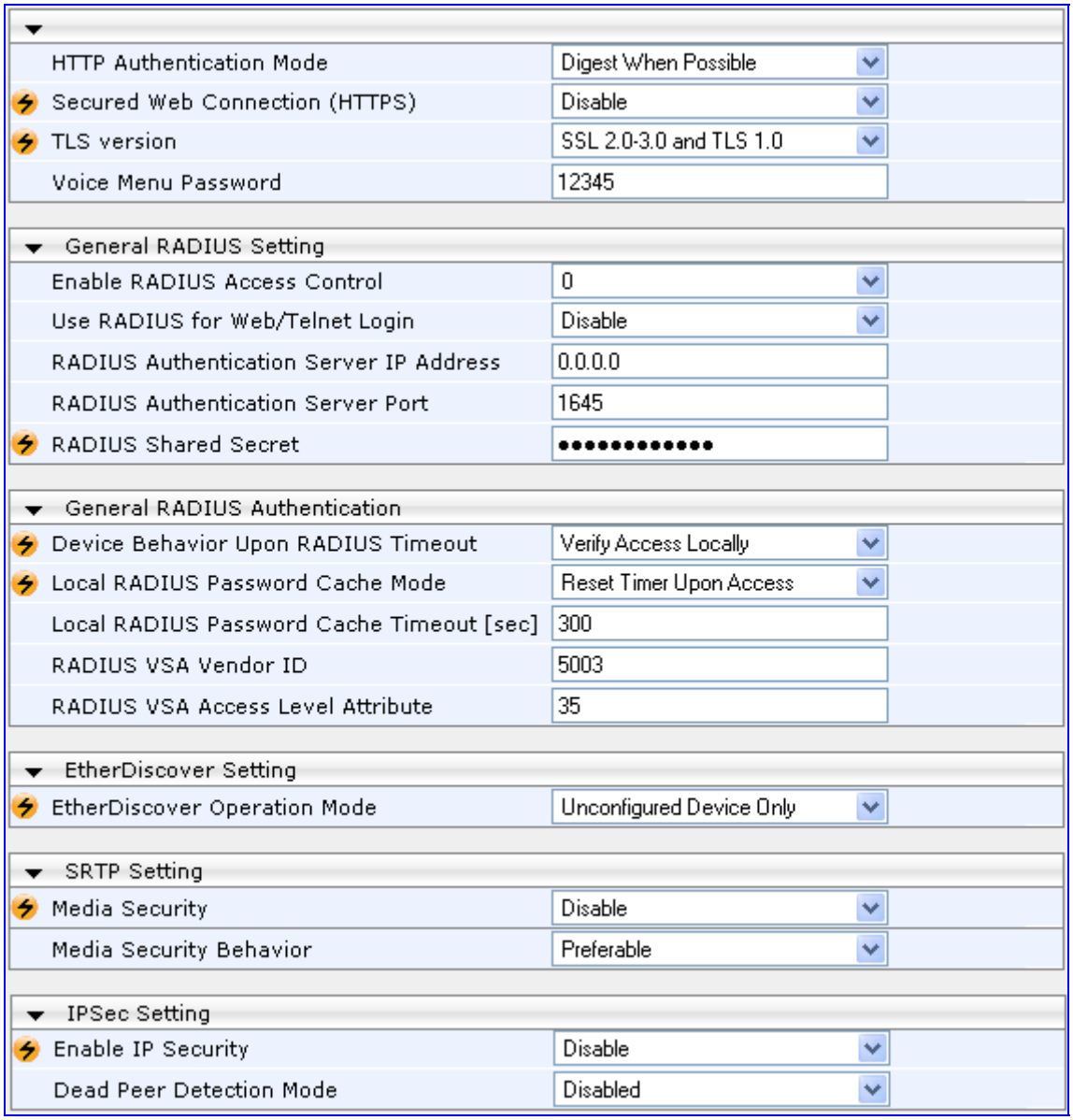
3.4.4.5 Configuring the General Security Settings

The 'General Security Settings' page is used to configure various security features.

➢ To configure the general security parameters, take these 4 steps:

1. Open the 'General Security Settings' page (**Configuration** tab > **Security Settings** menu > **General Security Settings** page item).

Figure 3-62: General Security Settings Page



2. Configure the General Security parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-26: General Security Parameters

Parameter	Description
HTTP Authentication Mode **[WebAuthMode]**	Determines the authentication mode for the Web interface. ▪ **[0]** Basic Mode = Basic authentication (clear text) is used (default). ▪ **[1]** Digest When Possible = Digest authentication (MD5) is used. ▪ **[2]** Basic if HTTPS, Digest if HTTP = Digest authentication (MD5) is used for HTTP, and basic authentication is used for HTTPS. **Note:** When RADIUS login is enabled (i.e., the parameter WebRADIUSLogin is set to 1), basic authentication is forced.
Secured Web Connection (HTTPS) **[HTTPSOnly]**	Determines the protocol types used to access the Web interface. ▪ **[0]** Disable = HTTP and HTTPS (default). ▪ **[1]** Enable = Unencrypted HTTP packets are blocked.
TLS version **[TLSVersion]**	Defines the supported versions of SSL/TLS (Secure Socket Layer/Transport Layer Security. ▪ **[0]** SSL 2.0-3.0 and TLS 1.0 = SSL 2.0, SSL 3.0, and TLS 1.0 are supported (default). ▪ **[1]** TLS 1.0 Only = only TLS 1.0 is used. When set to 0, SSL/TLS handshakes always start with SSL 2.0 and switch to TLS 1.0 if both peers support it. When set to 1, TLS 1.0 is the only version supported; clients attempting to contact the device using SSL 2.0 are rejected.
Voice Menu Password **[VoiceMenuPassword]**	Password for the voice menu used for configuration and status. To activate the menu, connect a POTS telephone and dial *** (three stars) followed by the password. The default value is 12345. For detailed information on the voice menu, refer to the device's Fast Track Guide. **Note:** This parameter is applicable only to Mediant 1000.
General RADIUS Settings	
Enable RADIUS Access Control **[EnableRADIUS]**	Determines whether the RADIUS application is enabled. ▪ **[0]** Disable = RADIUS application is disabled (default). ▪ **[1]** Enable = RADIUS application is enabled.
Use RADIUS for Web/Telnet Login **[WebRADIUSLogin]**	Uses RADIUS queries for Web and Telnet interface authentication. ▪ **[0]** Disable (default). ▪ **[1]** Enable. When enabled, logging in to the device's Web and Telnet embedded servers is performed via a RADIUS server. The device contacts a predefined server and verifies the given user name and password pair against a remote database, in a secure manner. **Notes:** ▪ The parameter EnableRADIUS must be set to 1. ▪ RADIUS authentication requires HTTP basic authentication, meaning the user name and password are transmitted in clear text over the network. Therefore, it's recommended to set the parameter HttpsOnly to 1 to force the use of HTTPS, since the transport is encrypted.

Parameter	Description
RADIUS Authentication Server IP Address **[RADIUSAuthServerIP]**	IP address of the RADIUS authentication server.
RADIUS Authentication Server Port **[RADIUSAuthPort]**	Port number of the RADIUS authentication server. The default value is 1645.
RADIUS Shared Secret **[SharedSecret]**	'Secret' used to authenticate the device to the RADIUS server. Should be a cryptographically strong password.
General RADIUS Authentication	
Device Behavior Upon RADIUS Timeout **[BehaviorUponRadiusTimeout]**	Defines device behavior upon a RADIUS timeout. ▪ **[0]** Deny Access = Denies access. ▪ **[1]** Verify Access Locally = Checks password locally (default).
Default Access Level **[DefaultAccessLevel]**	Defines the default access level for the device when the RADIUS (authentication) response doesn't include an access level attribute. The valid range is 0 to 255. The default value is 200 (Security Administrator').
Local RADIUS Password Cache Mode **[RadiusLocalCacheMode]**	Defines the device's mode of operation regarding the timer (configured by the parameter RadiusLocalCacheTimeout) that determines the validity of the user name and password (verified by the RADIUS server). ▪ **[0]** Absolute Expiry Timer = when you access a Web page, the timeout doesn't reset but instead, continues decreasing. ▪ **[1]** Reset Timer Upon Access = upon each access to a Web page, the timeout always resets (reverts to the initial value configured by RadiusLocalCacheTimeout).
Local RADIUS Password Cache Timeout **[RadiusLocalCacheTimeout]**	Defines the time (in seconds) the locally stored user name and password (verified by the RADIUS server) are valid. When this time expires, the user name and password become invalid and a must be re-verified with the RADIUS server. The valid range is 1 to 0xFFFFFF. The default value is 300 (5 minutes). ▪ **[-1]** = Never expires. ▪ **[0]** = Each request requires RADIUS authentication.
RADIUS VSA Vendor ID **[RadiusVSAVendorID]**	Defines the vendor ID that the device accepts when parsing a RADIUS response packet. The valid range is 0 to 0xFFFFFFFF. The default value is 5003.
RADIUS VSA Access Level Attribute **[RadiusVSAAccessAttribute]**	Defines the code that indicates the access level attribute in the Vendor Specific Attributes (VSA) section of the received RADIUS packet. The valid range is 0 to 255. The default value is 35.
EtherDiscover Setting	
EtherDiscover Operation Mode	N/A.

Parameter	Description
SRTP Setting	
Media Security **[EnableMediaSecurity]**	Enables Secure Real-Time Transport Protocol (SRTP). ▪ **[0]** Disable = SRTP is disabled (default). ▪ **[1]** Enable = SRTP is enabled.
Media Security Behavior **[MediaSecurityBehaviour]**	Determines the device's mode of operation when SRTP is used (EnableMediaSecurity = 1). ▪ **[0]** Preferable = The device initiates encrypted calls. If negotiation of the cipher suite fails, an unencrypted call is established. Incoming calls that don't include encryption information are accepted. ▪ **[1]** Mandatory = The device initiates encrypted calls, but if negotiation of the cipher suite fails, the call is terminated. Incoming calls that don't include encryption information are rejected (default).
IPSec Setting	
Enable IP Security **[EnableIPSec]**	Enables / disables the Internet Protocol security (IPSec) on the device. ▪ **[0]** Disable = IPSec is disabled (default). ▪ **[1]** Enable = IPSec is enabled.
Dead Peer Detection Mode **[IPSecDPDMode]**	Enables the Dead Peer Detection (DPD) 'keep-alive' mechanism (according to RFC 3706) to detect loss of peer connectivity. ▪ **[0]** Disabled (default). ▪ **[1]** Periodic = message exchanges at regular intervals. ▪ **[2]** On Demand = message exchanges as needed (i.e., before sending data to the peer). If the liveliness of the peer is questionable, the device sends a DPD message to query the status of the peer. If the device has no traffic to send, it never sends a DPD message. For detailed information on DPD, refer to the *Product Reference Manual*.

3.4.4.6 Configuring the IPSec Table

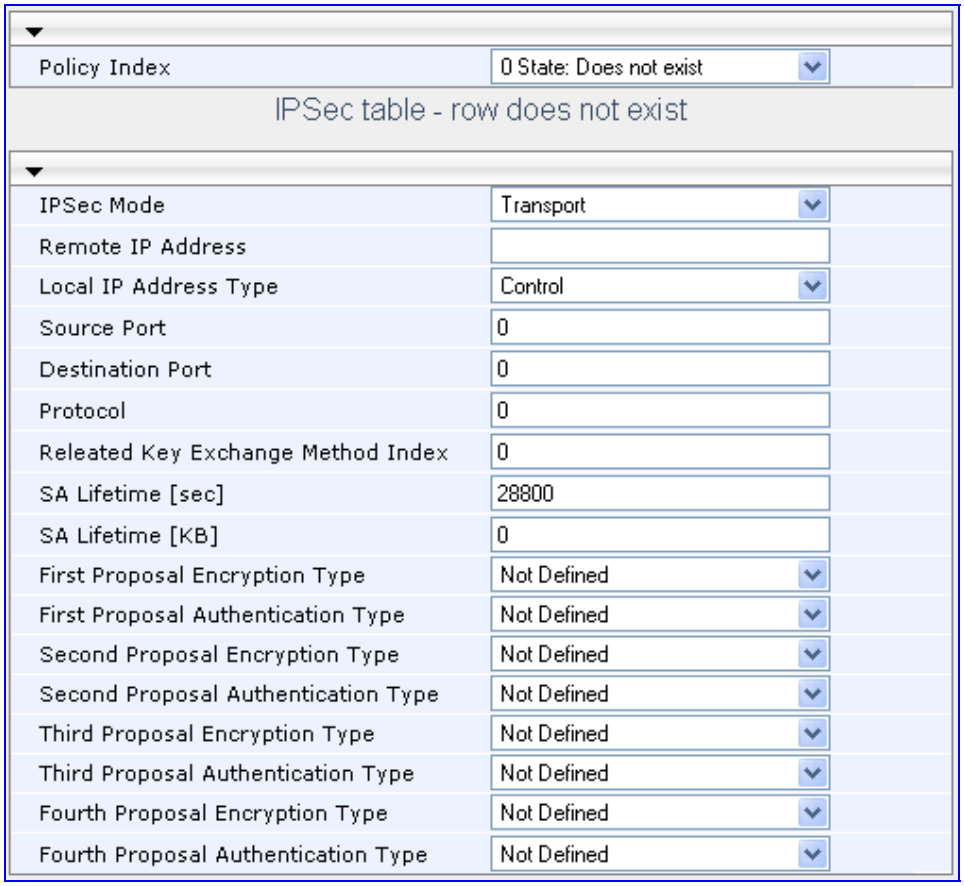
The 'IPSec Table' page allows you to configure the Security Policy Database (SPD) parameters for IP security (IPSec).

Note: You can also configure the IPSec table using the *ini* file table parameter IPSEC_SPD_TABLE (refer to "Security Parameters" on page 314).

➢ To configure the IPSec SPD table, take these 5 steps:

1. Open the 'IPSec Table' page (**Configuration** tab > **Security Settings** menu > **IPSec Table** page item).

Figure 3-63: IPSec Table Page



2. From the 'Policy Index' drop-down list, select the rule you want to edit (up to 20 policy rules can be configured).
3. Configure the IPSec SPD parameters according to the table below.
4. Click the button **Create**; the IPSec rule is applied on-the-fly to the device.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

If no IPSec methods are defined (Encryption / Authentication), the default settings, shown in the following table are applied.

Table 3-27: Default IKE Second Phase Proposals

Proposal	Encryption	Authentication
Proposal 0	3DES	SHA1
Proposal 1	3DES	MD5
Proposal 2	DES	SHA1
Proposal 3	DES	MD5

Table 3-28: IPSec SPD Table Configuration Parameters

Parameter Name	Description	
IPSec Mode **[IPSecMode]**	Defines the IPSec mode of operation. ▪ **[0]** Transport (Default) ▪ **[1]** Tunneling	
Remote Tunnel IP Address **[IPSecPolicyRemoteTunnelIPAddress]**	Defines the IP address of the remote IPSec tunneling device. **Note:** This parameter is only available if the parameter IPSecMode is set to Tunneling (1).	IPSec is applied to outgoing packets whose IP address, destination port, source port, and protocol type match the values defined for these parameters.
Remote Subnet Mask **[IPsecPolicyRemoteSubnetMask]**	Defines the subnet mask of the remote IPSec tunneling device. The default value is 255.255.255.255 (i.e., host-to-host IPSec tunnel). **Note:** This parameter is only available if the parameter IPSecMode is set to Tunneling (1).	
Remote IP Address **[IPSecPolicyRemoteIPAddress]**	Destination IP address (or FQDN) to which the IPSec mechanism is applied. **Notes:** ▪ This parameter is mandatory. ▪ When an FQDN is used, a DNS server must be configured (DNSPriServerIP).	
Local IP Address Type **[IPSecPolicyLocalIPAddressType]**	Determines the local interface to which the encryption is applied (applicable to multiple IPs and VLANs). ▪ **[0]** OAM = OAMP interface (default). ▪ **[1]** Control = Control interface.	
Source Port **[IPSecPolicySrcPort]**	Defines the source port to which the IPSec mechanism is applied. The default value is 0 (i.e., any port).	
Destination Port **[IPSecPolicyDstPort]**	Defines the destination port to which the IPSec mechanism is applied. The default value is 0 (i.e., any port).	
Protocol **[IPSecPolicyProtocol]**	Defines the protocol type to which the IPSec mechanism is applied. ▪ 0 = Any protocol (default). ▪ 17 = UDP. ▪ 6 = TCP. ▪ Any other protocol type defined by IANA (Internet Assigned Numbers Authority).	
Related Key Exchange Method Index **[IPsecPolicyKeyExchangeMethodIndex]**	Determines the index for the corresponding IKE entry. Note that several policies can be associated with a single IKE entry. The valid range is 0 to 19. The default value is 0.	

Parameter Name	Description
IKE Second Phase Parameters (Quick Mode)	
SA Lifetime (sec) **[PsecPolicyLifeInSec]**	Determines the time (in seconds) that the SA negotiated in the second IKE session (quick mode) is valid. After the time expires, the SA is re-negotiated. The default value is 28,800 (i.e., 8 hours).
SA Lifetime (KB) **[IPSecPolicyLifeInKB]**	Determines the lifetime (in kilobytes) that the SA negotiated in the second IKE session (quick mode) is valid. After this size is reached, the SA is re-negotiated. The default value is 0 (i.e., this parameter is ignored).
These lifetime parameters [SA Lifetime (sec) and SA Lifetime (KB)] determine the duration for which an SA is valid. When the lifetime of the SA expires, it is automatically renewed by performing the IKE second phase negotiations. To refrain from a situation where the SA expires, a new SA is negotiated while the old one is still valid. As soon as the new SA is created, it replaces the old one. This procedure occurs whenever an SA is about to expire.	
First to Fourth Proposal Encryption Type **[IPSecPolicyProposalEncryption_X]**	Determines the encryption type used in the quick mode negotiation for up to four proposals. For the *ini* file parameter, *X* depicts the proposal number (0 to 3)). The valid encryption values are: ▪ **[0]** None = No encryption ▪ **[1]** DES-CBC ▪ **[2]** Triple DES-CBC ▪ **[3]** AES-CBC ▪ Not Defined (default)
First to Fourth Proposal Authentication Type **[IPSecPolicyProposalAuthentication_X]**	Determines the authentication protocol used in the quick mode negotiation for up to four proposals. For the *ini* file parameter, *X* depicts the proposal number (0 to 3). The valid authentication values are: ▪ **[2]** HMAC-SHA-1-96 ▪ **[4]** HMAC-MD5-96 ▪ Not Defined (default)

3.4.4.7 Configuring the IKE Table

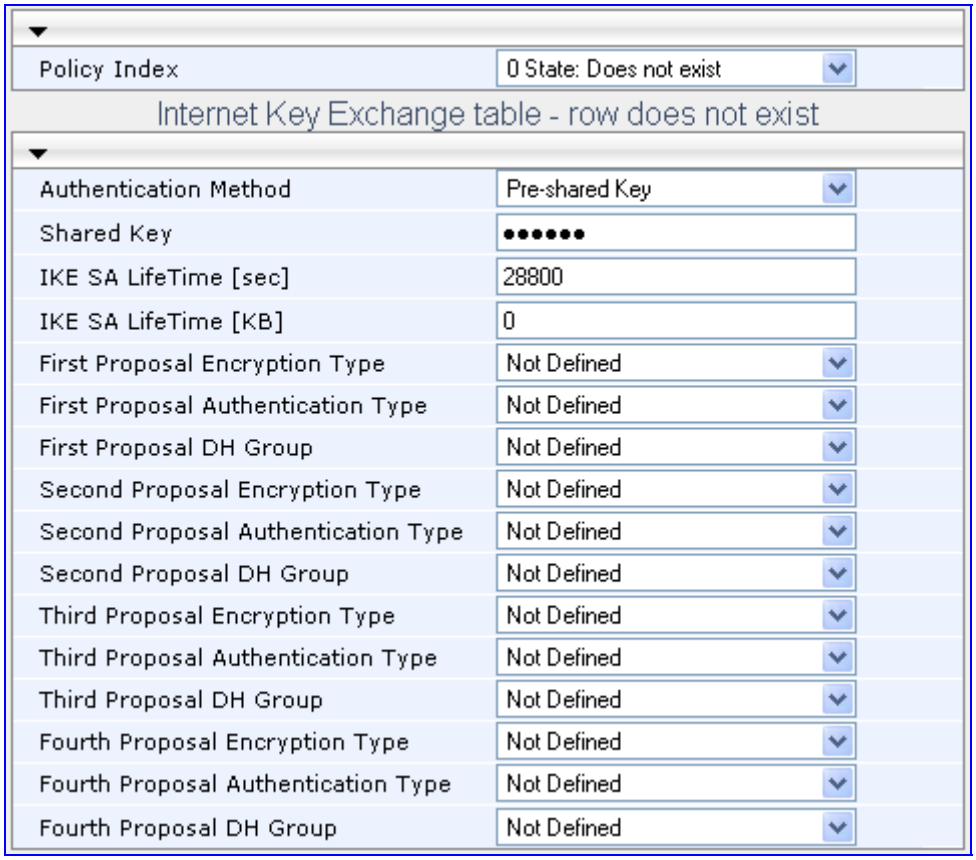
The 'IKE Table' page is used to configure the Internet Key Exchange (IKE) parameters.

Note: You can also configure the IKE table using the *ini* file table parameter IPSec_IKEDB_Table (refer to "Security Parameters" on page 314).

➢ To configure the IKE table, take these 5 steps:

1. Open the 'IKE Table' page (**Configuration** tab > **Security Settings** menu > **IKE Table** page item).

Figure 3-64: IKE Table Page



2. From the 'Policy Index' drop-down list, select the peer you want to edit (up to 20 peers can be configured).
3. Configure the IKE parameters according to the table below. Up to two IKE main mode proposals (Encryption / Authentication / DH group combinations) can be defined. The same proposals must be configured for all peers.
4. Click **Create**; a row is created in the IKE table.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

To delete a peer from the IKE table, select it from the 'Policy Index' drop-down list, click the button **Delete**, and then click **OK** at the prompt.

If no IKE methods are defined (Encryption / Authentication / DH Group), the default settings (shown in the following table) are applied.

Table 3-29: Default IKE First Phase Proposals

Proposal	Encryption	Authentication	DH Group
Proposal 0	3DES	SHA1	1024
Proposal 1	3DES	MD5	1024
Proposal 2	3DES	SHA1	786
Proposal 3	3DES	MD5	786

The parameters described in the following table are used to configure the first phase (main mode) of the IKE negotiation for a specific peer. A different set of parameters can be configured for each of the 20 available peers.

Table 3-30: IKE Table Configuration Parameters

Parameter Name	Description
Authentication Method	Determines the authentication method for IKE. ▪ **[0]** Pre-shared Key (default) ▪ **[1]** RSA Signature **Notes:** ▪ For pre-shared key authentication, peers participating in an IKE exchange must have a prior (out-of-band) knowledge of the common key (see IKEPolicySharedKey parameter). ▪ For RSA signature authentication, peers must be loaded with a certificate signed by a common CA. For additional information on certificates, refer to "Server Certificate Replacement" on page 118.
Shared Key	Determines the pre-shared key (in textual format). Both peers must register the same pre-shared key for the authentication process to succeed. **Notes:** ▪ The pre-shared key forms the basis of IPSec security and should therefore, be handled cautiously (in the same way as sensitive passwords). It is not recommended to use the same pre-shared key for several connections. ▪ Since the *ini* file is in plain text format, loading it to the device over a secure network connection is recommended, preferably over a direct crossed-cable connection from a management PC. For added confidentiality, use the encoded *ini* file option (described in "Secured Encoded ini File" on page 291). ▪ After it is configured, the value of the pre-shared key cannot be obtained via Web interface, *ini* file, or SNMP (refer the *Product Reference Manual*).
IKE SA LifeTime (sec)	Determines the time (in seconds) the SA negotiated in the first IKE session (main mode) is valid. After the time expires, the SA is re-negotiated. The default value is 28800 (i.e., 8 hours).

Parameter Name	Description
IKE SA LifeTime (KB)	Determines the lifetime (in kilobytes) that the SA negotiated in the first IKE session (main mode) is valid. After this size is reached, the SA is re-negotiated. The default value is 0 (i.e., this parameter is ignored).
These lifetime parameters [IKE SA LifeTime (sec) and IKE SA LifeTime (KB)] determine the duration the SA created in the main mode phase is valid. When the lifetime of the SA expires, it's automatically renewed by performing the IKE first phase negotiations. To refrain from a situation where the SA expires, a new SA is negotiated while the old one is still valid. As soon as the new SA is created, it replaces the old one. This procedure occurs whenever an SA is about to expire.	
First to Fourth Proposal Encryption Type	Determines the encryption type used in the main mode negotiation for up to four proposals. For the *ini* file parameter, *X* depicts the proposal number (0 to 3). ▪ **[1]** DES-CBC ▪ **[2]** Triple DES-CBC ▪ **[3]** AES-CBC ▪ Not Defined (default)
First to Fourth Proposal Authentication Type	Determines the authentication protocol used in the main mode negotiation for up to four proposals. For the *ini* file parameter, *X* depicts the proposal number (0 to 3). ▪ **[2]** HMAC-SHA1-96) ▪ **[4]** HMAC-MD5-96 ▪ Not Defined (default)
First to Fourth Proposal DH Group	Determines the length of the key created by the DH protocol for up to four proposals. For the *ini* file parameter, *X* depicts the proposal number (0 to 3). ▪ **[0]** DH-786-Bit ▪ **[1]** DH-1024-Bit ▪ Not Defined (default)

3.4.5 Protocol Configuration

The **Protocol Configuration** menu allows you to configure the device's SIP parameters and contains the following submenus:

- Protocol Definition (refer to "Configuring the Protocol Definition Parameters" on page 133)
- SIP Advanced Parameters (refer to "Configuring the SIP Advanced Parameters" on page 165)
- Manipulation Tables (refer to "Configuring the Number Manipulation Tables" on page 190)
- Routing Tables (refer to "Configuring the Routing Tables" on page 197)
- Profile Definitions (refer to "Configuring the Profile Definitions" on page 212)
- Endpoint Settings (refer to "Configuring the Endpoint Settings" on page 218)
- Trunk/IP Group (refer to "Configuring the Trunk and IP Groups" on page 226)
- Digital Gateway (refer to "Configuring the Digital Gateway Parameters" on page 235)
- IP Media (refer to "Configuring the IPmedia Parameters" on page 242)

3.4.5.1 Configuring the Protocol Definition Parameters

The **Protocol Definition** submenu allows you to configure the main SIP protocol parameters. This submenu contains the following page items:

- SIP General Parameters (refer to "SIP General Parameters" on page 134)
- Proxy & Registration (refer to "Proxy & Registration Parameters" on page 146)
- Proxy Sets Table (refer to "Proxy Sets Table" on page 153)
- Coders (refer to "Coders" on page 157)
- DTMF & Dialing (refer to "DTMF & Dialing Parameters" on page 160)

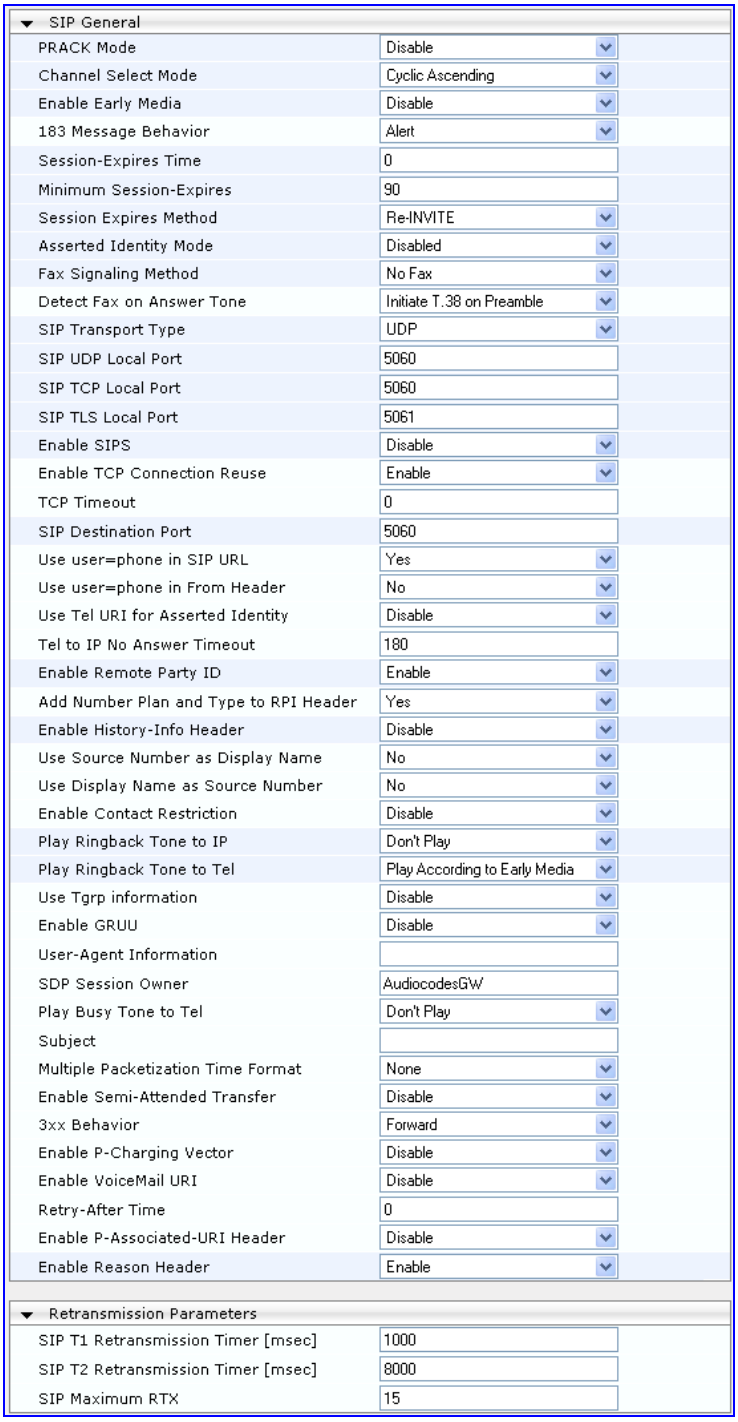
3.4.5.1.1 SIP General Parameters

The 'SIP General Parameters' page is used to configure general SIP parameters.

➢ To configure the general SIP protocol parameters, take these 4 steps:

1. Open the 'SIP General Parameters' page (**Configuration** tab > **Protocol Configuration** menu > **Protocol Definition** submenu > **SIP General Parameters** page item).

Figure 3-65: SIP General Parameters


SIP General
PRACK Mode: Disable
Channel Select Mode: Cyclic Ascending
Enable Early Media: Disable
183 Message Behavior: Alert
Session-Expires Time: 0
Minimum Session-Expires: 90
Session Expires Method: Re-INVITE
Asserted Identity Mode: Disabled
Fax Signaling Method: No Fax
Detect Fax on Answer Tone: Initiate T.38 on Preamble
SIP Transport Type: UDP
SIP UDP Local Port: 5060
SIP TCP Local Port: 5060
SIP TLS Local Port: 5061
Enable SIPS: Disable
Enable TCP Connection Reuse: Enable
TCP Timeout: 0
SIP Destination Port: 5060
Use user=phone in SIP URL: Yes
Use user=phone in From Header: No
Use Tel URI for Asserted Identity: Disable
Tel to IP No Answer Timeout: 180
Enable Remote Party ID: Enable
Add Number Plan and Type to RPI Header: Yes
Enable History-Info Header: Disable
Use Source Number as Display Name: No
Use Display Name as Source Number: No
Enable Contact Restriction: Disable
Play Ringback Tone to IP: Don't Play
Play Ringback Tone to Tel: Play According to Early Media
Use Tgrp information: Disable
Enable GRUU: Disable
User-Agent Information:
SDP Session Owner: AudiocodesGW
Play Busy Tone to Tel: Don't Play
Subject:
Multiple Packetization Time Format: None
Enable Semi-Attended Transfer: Disable
3xx Behavior: Forward
Enable P-Charging Vector: Disable
Enable VoiceMail URI: Disable
Retry-After Time: 0
Enable P-Associated-URI Header: Disable
Enable Reason Header: Enable
Retransmission Parameters
SIP T1 Retransmission Timer [msec]: 1000
SIP T2 Retransmission Timer [msec]: 8000
SIP Maximum RTX: 15

2. Configure the parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-31: SIP General Parameters (Protocol Definition)

Parameter	Description
PRACK Mode **[PRACKMode]**	PRACK (Provisional Acknowledgment) mechanism mode for 1xx SIP reliable responses. ▪ **[0]** Disable ▪ **[1]** Supported (default) ▪ **[2]** Required **Notes:** ▪ The Supported and Required headers contain the '100rel' tag. ▪ The device sends PRACK messages if the 180/183 response is received with '100rel' in the Supported or Required headers.

Parameter	Description
Channel Select Mode **[ChannelSelectMode]**	Port (channel) allocation algorithm for IP-to-Tel calls. ▪ **[0]** By Dest Phone Number = Selects the device's channel according to the called number (defined in the 'Endpoint Phone Number' table -- Configuring the Endpoint Phone Numbers). (default.) ▪ **[1]** Cyclic Ascending = Selects the next available channel in an ascending cyclic order. Always selects the next higher channel number in the trunk group. When the device reaches the highest channel number in the trunk group, it selects the lowest channel number in the trunk group and then starts ascending again. ▪ **[2]** Ascending = Selects the lowest available channel. It always starts at the lowest channel number in the trunk group and if that channel is not available, selects the next higher channel. ▪ **[3]** Cyclic Descending = Selects the next available channel in descending cyclic order. Always selects the next lower channel number in the trunk group. When the device reaches the lowest channel number in the hunt group, it selects the highest channel number in the trunk group and then starts descending again. ▪ **[4]** Descending = Selects the highest available channel. Always starts at the highest channel number in the trunk group and if that channel is not available, selects the next lower channel. ▪ **[5]** Dest Number + Cyclic Ascending = First selects the device's port according to the called number. If the called number isn't found, it then selects the next available channel in ascending cyclic order. Note that if the called number is found, but the port associated with this number is busy, the call is released. ▪ **[6]** By Source Phone Number = Selects the device's channel according to the calling number. ▪ [7] Trunk Cyclic Ascending = Digital: Selects the device's port from the first channel of the next trunk (next to the trunk from which the previous channel was allocated. Not applicable for Mediant 1000 analog. **Notes:** ▪ The internal numbers of the device's B-channels are defined by the TrunkGroup parameter. ▪ For defining the channel select mode per Trunk Group, refer to "Configuring the Trunk Group Settings" on page 228.
Enable Early Media **[EnableEarlyMedia]**	Enables the device to send a 183 Session Progress response with SDP (instead of 180 Ringing), allowing the media stream to be established prior to the answering of the call. ▪ **[0]** Disable = Early Media is disabled (default). ▪ **[1]** Enable = Enables Early Media. For Mediant 1000 Analog interface: Note that to send a 183 response, you must also set the parameter ProgressIndicator2IP to 1. If it is equal to 0, 180 Ringing response is sent. For Digital interface: Sending a 183 response depends on the Progress Indicator (PI). It is sent only if PI is set to 1 or 8 are received in Proceeding or Alert PRI messages. For CAS devices, see the ProgressIndicator2IP parameter.

Parameter	Description
183 Message Behavior **[SIP183Behaviour]**	Defines the ISDN message that is sent when the 183 Session Progress message is received for IP-to-Tel calls. ▪ **[0]** Progress = The device sends a PROGRESS message (default). ▪ **[1]** Alert = The device sends an ALERT message (upon receipt of a 183 response) instead of an ISDN PROGRESS message.
Session-Expires Time **[SIPSessionExpires]**	Determines the numerical value that is sent in the Session-Expires header in the first INVITE request or response (if the call is answered). The valid range is 1 to 86,400 sec. The default is 0 (i.e., the Session-Expires header is disabled).
Minimum Session-Expires **[MinSE]**	Defines the time (in seconds) that is used in the Min-SE header. This header defines the minimum time that the user agent refreshes the session. The valid range is 10 to 100,000. The default value is 90.
Session Expires Method **[SessionExpiresMethod]**	Determines the SIP method used for session-timer updates. ▪ **[0]** Re-INVITE = Uses Re-INVITE messages for session-timer updates (default). ▪ **[1]** UPDATE = Uses UPDATE messages. **Notes:** ▪ The device can receive session-timer refreshes using both methods. ▪ The UPDATE message used for session-timer is excluded from the SDP body.
Asserted Identity Mode **[AssertedIdMode]**	Determines whether P-Asserted-Identity or P-Preferred-Identity is used in the generated INVITE request for Caller ID (or privacy). ▪ **[0]** Disabled = None (default) ▪ **[1]** Adding PAsserted Identity ▪ **[2]** Adding PPreferred Identity The Asserted ID mode defines the header (P-Asserted-Identity or P-Preferred-Identity) that is used in the generated INVITE request. The header also depends on the calling Privacy (allowed or restricted). The P-Asserted-Identity (or P-Preferred-Identity) headers are used to present the originating party's Caller ID. The Caller ID is composed of a Calling Number and (optionally) a Calling Name. P-Asserted-Identity (or P-Preferred-Identity) headers are used together with the Privacy header. If Caller ID is restricted (P-Asserted-Identity is not sent), the Privacy header includes the value 'id' ('Privacy: id'). Otherwise, for allowed Caller ID, 'Privacy: none' is used. If Caller ID is restricted (received from Tel or configured in the device), the From header is set to <anonymous@anonymous.invalid>. The logic for filling the calling party parameters is as follows: the SIP header is selected first from which the calling party parameters are obtained: first priority is P-Asserted-Identity, second is Remote-Party-ID, and third is the From header. Once a URL is selected, all the calling party parameters are set from this header. If P-Asserted-Identity is selected, the Privacy header is checked and if the Privacy is set to 'id', the calling number is assumed restricted.

Parameter	Description
Fax Signaling Method **[IsFaxUsed]**	Determines the SIP signaling method for establishing and transmitting a fax session after a fax is detected. ▪ **[0]** No Fax = No fax negotiation using SIP signaling. Fax transport method is according to the parameter FaxTransportMode (default). ▪ **[1]** T.38 Relay = Initiates T.38 fax relay. ▪ **[2]** G.711 Transport = Initiates fax / modem using the coder G.711 A-law/μ-law with adaptations (refer to Note below). ▪ **[3]** Fax Fallback = Initiates T.38 fax relay. If the T.38 negotiation fails, the device re-initiates a fax session using the coder G.711 A-law/μ-law with adaptations (refer to the Note below). **Notes:** ▪ Fax adaptations (for options 2 and 3): Echo Canceller = On Silence Compression = Off Echo Canceller Non-Linear Processor Mode = Off Dynamic Jitter Buffer Minimum Delay = 40 Dynamic Jitter Buffer Optimization Factor = 13 ▪ If the device initiates a fax session using G.711 (option 2 and possibly 3), a 'gpmd' attribute is added to the SDP in the following format: For A-law: 'a=gpmd:8 vbd=yes;ecan=on'. For μ-law: 'a=gpmd:0 vbd=yes;ecan=on'. ▪ When IsFaxUsed is set to 1, 2, or 3, the parameter FaxTransportMode is ignored. ▪ When the value of IsFaxUsed is other than 1, T.38 might still be used without the control protocol's involvement. To completely disable T.38, set FaxTransportMode to a value other than 1. ▪ For detailed information on fax transport methods, refer to "Fax/Modem Transport Modes" on page 399.
Detect Fax on Answer Tone **[DetFaxOnAnswerTone]**	Determines when the device initiates a T.38 session for fax transmission. ▪ **[0]** Initiate T.38 on Preamble = device to which terminating fax is connected initiates T.38 session on receiving HDLC preamble signal from fax (default). ▪ **[1]** Initiate T.38 on CED = device to which terminating fax is connected initiates T.38 session on receiving CED answer tone from fax. **Note:** This parameters is applicable only if IsFaxUsed is set to 1.
SIP Transport Type **[SIPTransportType]**	Determines the default transport layer for outgoing SIP calls initiated by the device. ▪ **[0]** UDP (default) ▪ **[1]** TCP ▪ **[2]** TLS (SIPS) **Notes:** ▪ It's recommended to use TLS for communication with a SIP Proxy and not for direct device-to-device communication. ▪ The value of this parameter is also used by the SAS application as the default transport layer for outgoing SIP calls - only applicable for Mediant 1000.
SIP UDP Local Port **[LocalSIPPort]**	Local UDP port for SIP messages. The valid range is 1 to 65534. The default value is 5060.

Parameter	Description
SIP TCP Local Port **[TCPLocalSIPPort]**	Local TCP port for SIP messages. The valid range is 1 to 65534. The default value is 5060.
SIP TLS Local Port **[TLSLocalSIPPort]**	Local TLS port for SIP messages. The valid range is 1 to 65534. The default value is 5061. **Note:** The value of must be different than the value of 'SIP TCP Local Port' (TCPLocalSIPPort).
Enable SIPS **[EnableSIPS]**	Enables secured SIP (SIPS URI) connections over multiple hops. ▪ **[0]** Disable (default). ▪ **[1]** Enable. When 'SIP Transport Type' is set to TLS (SIPTransportType = 2) and 'Enable SIPS' is disabled, TLS is used for the next network hop only. When 'SIP Transport Type' is set to TCP or TLS (SIPTransportType = 2 or 1) and 'Enable SIPS' is enabled, TLS is used through the entire connection (over multiple hops). **Note:** If this parameter is enabled and 'SIP Transport Type' is set to UDP (SIPTransportType = 0), the connection fails.
Enable TCP Connection Reuse **[EnableTCPConnectionReuse]**	Enables the reuse of the same TCP connection for all calls to the same destination. ▪ **[0]** Disable = Use a separate TCP connection for each call (default). ▪ **[1]** Enable = Use the same TCP connection for all calls.
TCP Timeout **[SIPTCPTimeout]**	Defines the Timer B (INVITE transaction timeout timer) and Timer F (non-INVITE transaction timeout timer), as defined in RFC 3261, when the SIP Transport Type is TCP. The valid range is 0 to 40 sec. The default value is 64*SIPT1Rtx msec.
SIP Destination Port **[SIPDestinationPort]**	SIP destination port for sending initial SIP requests. The valid range is 1 to 65534. The default port is 5060. **Note:** SIP responses are sent to the port specified in the Via header.
Use user=phone in SIP URL **[IsUserPhone]**	Determines whether to add 'user=phone' string in SIP URI. ▪ **[0]** No = 'user=phone' string isn't used in SIP URI. ▪ **[1]** Yes = 'user=phone' string is part of the SIP URI (default).
Use user=phone in From Header **[IsUserPhoneInFrom]**	Determines whether to add 'user=phone' string in the From header. ▪ **[0]** No = Doesn't use 'user=phone' string in From header (default). ▪ **[1]** Yes = 'user=phone' string is part of the From header.
Use Tel URI for Asserted Identity **[UseTelURIForAssertedID]**	Determines the format of the URI in the P-Asserted-Identity and P-Preferred-Identity headers. ▪ **[0]** Disable = 'sip:' (default). ▪ **[1]** Enable = 'tel:'.
Tel to IP No Answer Timeout **[IPAlertTimeout]**	Defines the time (in seconds) that the device waits for a 200 OK response from the called party (IP side) after sending an INVITE message. If the timer expires, the call is released. The valid range is 0 to 3600. The default value is 180.
Enable Remote Party ID **[EnableRPIheader]**	Enables Remote-Party-ID (RPI) headers for calling and called numbers for Tel-to-IP calls. ▪ **[0]** Disable (default). ▪ **[1]** Enable = RPI headers are generated in SIP INVITE messages for both called and calling numbers.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Add Number Plan and Type to Remote Party ID Header
[AddTON2RPI]</td><td>Determines whether the TON/PLAN parameters are included in the Remote-Party-ID (RPID) header.
▪ [0] No
▪ [1] Yes (default)
If RPID header is enabled (EnableRPIHeader = 1) and AddTON2RPI = 1, it's possible to configure the calling and called number type and number plan using the Number Manipulation tables for Tel-to-IP calls.</td></tr>
<tr><td>Enable History-Info Header
[EnableHistoryInfo]</td><td>Enables usage of the History-Info header.
▪ [0] Disable = Disable (default)
▪ [1] Enable = Enable
User Agent Client (UAC) Behavior:
▪ Initial request: The History-Info header is equal to the Request URI. If a PSTN Redirect number is received, it is added as an additional History-Info header with an appropriate reason.
▪ Upon receiving the final failure response, the device copies the History-Info as is, adds the reason of the failure response to the last entry, and concatenates a new destination to it (if an additional request is sent). The order of the reasons is as follows:
1. Q.850 Reason
2. SIP Reason
3. SIP Response code
▪ Upon receiving the final response (success or failure), the device searches for a Redirect reason in the History-Info (i.e., 3xx/4xx SIP reason). If found, it is passed to ISDN according to the following table:
<table>
<tr><th>SIP Reason Code</th><th>ISDN Redirecting Reason</th></tr>
<tr><td>302 - Moved Temporarily</td><td>Call Forward Universal (CFU)</td></tr>
<tr><td>408 - Request Timeout</td><td rowspan="2">Call Forward No Answer (CFNA)</td></tr>
<tr><td>480 - Temporarily Unavailable</td></tr>
<tr><td>486 - Busy Here</td><td rowspan="2">Call Forward Busy (CFB)</td></tr>
<tr><td>600 - Busy Everywhere</td></tr>
</table>
▪ If history reason is a Q.850 reason, it is translated to the SIP reason (according to the SIP-ISDN tables) and then to ISDN Redirect reason according to the table above.
User Agent Server (UAS) Behavior:
▪ The History-Info header is sent only in the final response.
▪ Upon receiving a request with History-Info, the UAS checks the policy in the request. If 'session', 'header', or 'history' policy tag is found, the (final) response is sent without History-Info; otherwise, it is copied from the request.</td></tr>
</table>

Parameter	Description
Use Source Number as Display Name **[UseSourceNumberAsDisplayName]**	Determines the use of Tel Source Number and Display Name for Tel-to-IP calls. ▪ **[0]** No = If a Tel Display Name is received, the Tel Source Number is used as the IP Source Number and the Tel Display Name is used as the IP Display Name. If no Display Name is received from the Tel side, the IP Display Name remains empty (default). ▪ **[1]** Yes = If a Tel Display Name is received, the Tel Source Number is used as the IP Source Number and the Tel Display Name is used as the IP Display Name. If no Display Name is received from the Tel side, the Tel Source Number is used as the IP Source Number and also as the IP Display Name. ▪ **[2]** Overwrite = The Tel Source Number is used as the IP Source Number and also as the IP Display Name (even if the received Tel Display Name is not empty).
Use Display Name as Source Number **[UseDisplayNameAsSourceNumber]**	Determines the use of Source Number and Display Name for IP-to-Tel calls. ▪ **[0]** No = If IP Display Name is received, the IP Source Number is used as the Tel Source Number and the IP Display Name is used as the Tel Display Name. If no Display Name is received from IP, the Tel Display Name remains empty (default). ▪ **[1]** Yes = If an IP Display Name is received, it is used as the Tel Source Number and also as the Tel Display Name, and Presentation is set to Allowed (0). If no Display Name is received from IP, the IP Source Number is used as the Tel Source Number and Presentation is set to Restricted (1). For example: When 'from: 100 <sip:200@201.202.203.204>' is received, the outgoing Source Number and Display Name are set to '100' and the Presentation is set to Allowed (0). When 'from: <sip:100@101.102.103.104>' is received, the outgoing Source Number is set to '100' and the Presentation is set to Restricted (1).
Enable Contact Restriction **[EnableContactRestriction]**	Determines whether the device sets the Contact header of outgoing INVITE requests to ‘anonymous’ for restricted calls. ▪ **[0]** = Disabled (default) ▪ **[1]** = Enabled

Parameter	Description
Play Ringback Tone to IP **[PlayRBTone2IP]**	Determines whether or not the device plays a ringback tone (RBT) to the IP side of the call (IP-to-Tel calls). ▪ **[0]** Don't Play = Ringback tone isn't played (default). ▪ **[1]** Play = Ringback tone is played after SIP 183 session progress response is sent. For digital modules: If configured to 1 ('Play') and EnableEarlyMedia = 1, the device plays a ringback tone according to the following: ▪ For CAS interfaces: the device opens a voice channel, sends a 183+SDP response, and then plays a ringback tone to IP. ▪ For ISDN interfaces: if a Progress or an Alert message with PI (1 or 8) is received from the ISDN, the device opens a voice channel, sends a 183+SDP or 180+SDP response, but doesn't play a ringback tone to IP. If PI (1 or 8) is received from the ISDN, the device assumes that ringback tone is played by the ISDN switch. Otherwise, the device plays a ringback tone to IP after receiving an Alert message from the ISDN. It sends a 180+SDP response, signaling to the calling party to open a voice channel to hear the played ringback tone. **Notes:** ▪ This parameter is applicable only to Mediant 1000 FXS interfaces. ▪ To enable the device to send a 183/180+SDP responses, set EnableEarlyMedia to 1. ▪ If EnableDigitDelivery = 1, the device doesn't play a ringback tone to IP and doesn't send 183 or 180+SDP responses.
Play Ringback Tone to Tel **[PlayRBTone2Tel]**	▪ Determines the method used to play a ringback tone to the Tel side. **[0]** Don't Play = Ringback tone isn't played. ▪ **[1]** Play Local = Ringback tone is played to the Tel side of the call when 180/183 response is received. ▪ **[2]** Play According to Early Media = Ringback tone is played to the Tel side of the call if no SDP is received in 180/183 responses. If 180/183 with SDP message is received, the device cuts through the voice channel and doesn't play ringback tone (default).
Use Tgrp Information **[UseSIPTgrp]**	Determines whether the SIP 'tgrp' parameter, which specifies the Trunk Group to which the call belongs is used. ▪ **[0]** Disable = The 'tgrp' parameter isn't used (default). ▪ **[1]** Send Only = The Trunk Group number is added to the 'tgrp' parameter value in the Contact header of outgoing SIP messages. If a Trunk Group number is not associated with the call, the 'tgrp' parameter isn't included. If a 'tgrp' value is specified in incoming messages, it is ignored. ▪ **[2]** Send and Receive = The functionality of outgoing SIP messages is identical to the functionality described in option (1). In addition, for incoming SIP messages, if the Request-URI includes a 'tgrp' parameter, the device routes the call according to that value (if possible). If the Contact header includes a 'tgrp' parameter, it is copied to the corresponding outgoing messages in that dialog.

Parameter	Description
Enable GRUU **[EnableGRUU]**	Determines whether the Globally Routable User Agent URIs (GRUU) mechanism is used. ▪ **[0]** Disable = Disable (default) ▪ **[1]** Enable = Enable The device obtains a GRUU by generating a normal REGISTER request. This request contains a Supported header with the value 'gruu'. The device includes a '+sip.instance' Contact header parameter for each contact for which the GRUU is desired. This Contact parameter contains a globally unique ID that identifies the device instance. The global unique ID is as follows: ▪ If registration is per endpoint (AuthenticationMode=0), it is the MAC address of the device concatenated with the phone number of the endpoint. ▪ If the registration is per device (AuthenticationMode=1) it is only the MAC address. ▪ When the User Information mechanism is used, the globally unique ID is the MAC address concatenated with the phone number of the endpoint (defined in the User-Info file). If the Registrar/Proxy supports GRUU, the REGISTER responses contain the 'gruu' parameter in each Contact header field. The Registrar/Proxy provides the same GRUU for the same AOR and instance-id in case of sending REGISTER again after expiration of the registration. The device places the GRUU in any header field which contains a URI. It uses the GRUU in the following messages: INVITE requests, 2xx responses to INVITE, SUBSCRIBE requests, 2xx responses to SUBSCRIBE, NOTIFY requests, REFER requests, and 2xx responses to REFER. **Note:** If the GRUU contains the 'opaque' URI parameter, the device obtains the AOR for the user by stripping the parameter. The resulting URI is the AOR. For example: AOR: sip:alice@example.com GRUU: sip:alice@example.com;opaque="kjh29x97us97d"
User-Agent Information **[UserAgentDisplayInfo]**	Defines the string that is used in the SIP request header User-Agent and SIP response header Server. If not configured, the default string 'AudioCodes product-name s/w-version' is used (e.g., User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006). When configured, the string 'UserAgentDisplayInfo s/w-version' is used (e.g., User-Agent: MyNewOEM/v.5.40.010.006). Note that the version number can't be modified. The maximum string length is 50 characters.
SDP Session Owner **[SIPSDPSessionOwner]**	Determines the value of the Owner line ('o' field) in outgoing SDP messages. The valid range is a string of up to 39 characters. The default value is 'AudiocodesGW'. For example: o=AudiocodesGW 1145023829 1145023705 IN IP4 10.33.4.126

Parameter	Description
Play Busy Tone to Tel **[PlayBusyTone2ISDN]**	Enables the device to play a busy or reorder tone to the PSTN after a Tel-to-IP call is released. ▪ **[0]** Don't Play = Immediately sends an ISDN Disconnect message (default). ▪ **[1]** Play when Disconnecting = Sends an ISDN Disconnect message with PI = 8 and plays a busy or reorder tone to the PSTN (depending on the release cause). ▪ **[2]** Play before Disconnect = Delays the sending of an ISDN Disconnect message for a user-defined time (configured by the TimeForReorderTone parameter) and plays a busy or reorder tone to the PSTN. Applicable only if the call is released from the IP [Busy Here (486) or Not Found (404)] before it reaches the Connect state; otherwise, the Disconnect message is sent immediately and no tones are played.
Subject **[SIPSubject]**	Defines the value of the Subject header in outgoing INVITE messages. If not specified, the Subject header isn't included (default). The maximum length is up to 50 characters.
Multiple Packetization Time Format **[MultiPtimeFormat]**	Determines whether the 'mptime' attribute is included in the outgoing SDP. ▪ **[0]** None = Disabled (default) ▪ **[1]** PacketCable = includes the 'mptime' attribute in the outgoing SDP -- PacketCable-defined format The 'mptime' attribute enables the device to define a separate Packetization period for each negotiated coder in the SDP. The 'mptime' attribute is only included if this parameter is enabled, even if the remote side includes it in the SDP offer. Upon receipt, each coder receives its 'ptime' value in the following precedence: from 'mptime' attribute, from 'ptime' attribute, and then from default value.
Enable Reason Header **[EnableReasonHeader]**	Enables / disables the usage of the SIP Reason header. ▪ **[0]** Disable. ▪ **[1]** Enable (default).
Enable Semi-Attended Transfer **[EnableSemiAttendedTransfer]**	Determines the device behavior when Transfer is initiated while in Alerting state. ▪ **[0]** Disable = Send REFER with Replaces (default). ▪ **[1]** Enable = Send CANCEL, and after a 487 response is received, send REFER without Replaces.
3xx Behavior **[3xxBehavior]**	Determines the device's behavior regarding call identifiers when a 3xx response is received for an outgoing INVITE request. The device can either use the same call identifiers (Call-ID, Branch, To, and From tags) or change them in the new initiated INVITE. ▪ **[0]** Forward = Use different call identifiers for a redirected INVITE message (default). ▪ **[1]** Redirect = Use the same call identifiers.
Enable P-Charging Vector **[EnablePChargingVector]**	Enables the addition of a P-Charging-Vector header to all outgoing INVITE messages. ▪ **[0]** Disable = Disable (default) ▪ **[1]** Enable = Enable

Parameter	Description
Enable VoiceMail URI **[EnableVMURI]**	Enables or disables the interworking of target and cause for redirection from Tel to IP and vice versa, according to RFC 4468. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Enable = Enable Upon receipt of an ISDN SETUP message with redirect values, the device maps the Redirect phone number to the SIP 'target' parameter and the Redirect number reason to the SIP 'cause' parameter in the Request-URI. **Redirecting Reason >> SIP Response Code** Unknown >> 404 User busy >> 486 No reply >> 408 Deflection >> 487/480 Unconditional >> 302 Others >> 302 If the device receives a Request-URI that includes a 'target' and 'cause' parameter, the 'target' is mapped to the Redirect phone number and the 'cause' is mapped to Redirect number reason.
Retry-After Time **[RetryAfterTime]**	Determines the time (in seconds) used in the Retry-After header when a 503 (Service Unavailable) response is generated by the device. The time range is 0 to 3,600. The default value is 0.
Enable P-Associated-URI Header **[EnablePAssociatedURIHeader]**	Determines the device usage of the P-Associated-URI header. This header can be received in 200 OK responses to REGISTER requests. When enabled, the first URI in the P-Associated-URI header is used in subsequent requests as the From / P-Asserted-Id headers value. ▪ **[0]** Disable (default). ▪ **[1]** Enable. **Note:** P-Associated-URIs in registration responses is handled only if the device is registered per endpoint (using the User Information file).
Retransmission Parameters	
SIP T1 Retransmission Timer [msec] **[SipT1Rtx]**	The time interval (in msec) between the first transmission of a SIP message and the first retransmission of the same message. The default is 500. **Note:** The time interval between subsequent retransmissions of the same SIP message starts with SipT1Rtx and is multiplied by two until SipT2Rtx. For example (assuming that SipT1Rtx = 500 and SipT2Rtx = 4000): ▪ The first retransmission is sent after 500 msec. ▪ The second retransmission is sent after 1000 (2*500) msec. ▪ The third retransmission is sent after 2000 (2*1000) msec. ▪ The fourth retransmission and subsequent retransmissions until SIPMaxRtx are sent after 4000 (2*2000) msec.
SIP T2 Retransmission Timer [msec] **[SipT2Rtx]**	The maximum interval (in msec) between retransmissions of SIP messages. The default is 4000. **Note:** The time interval between subsequent retransmissions of the same SIP message starts with SipT1Rtx and is multiplied by two until SipT2Rtx.

Parameter	Description
SIP Maximum RTX **[SIPMaxRtx]**	Maximum number of UDP transmissions (first transmission plus retransmissions) of SIP messages. The range is 1 to 30. The default value is 7.

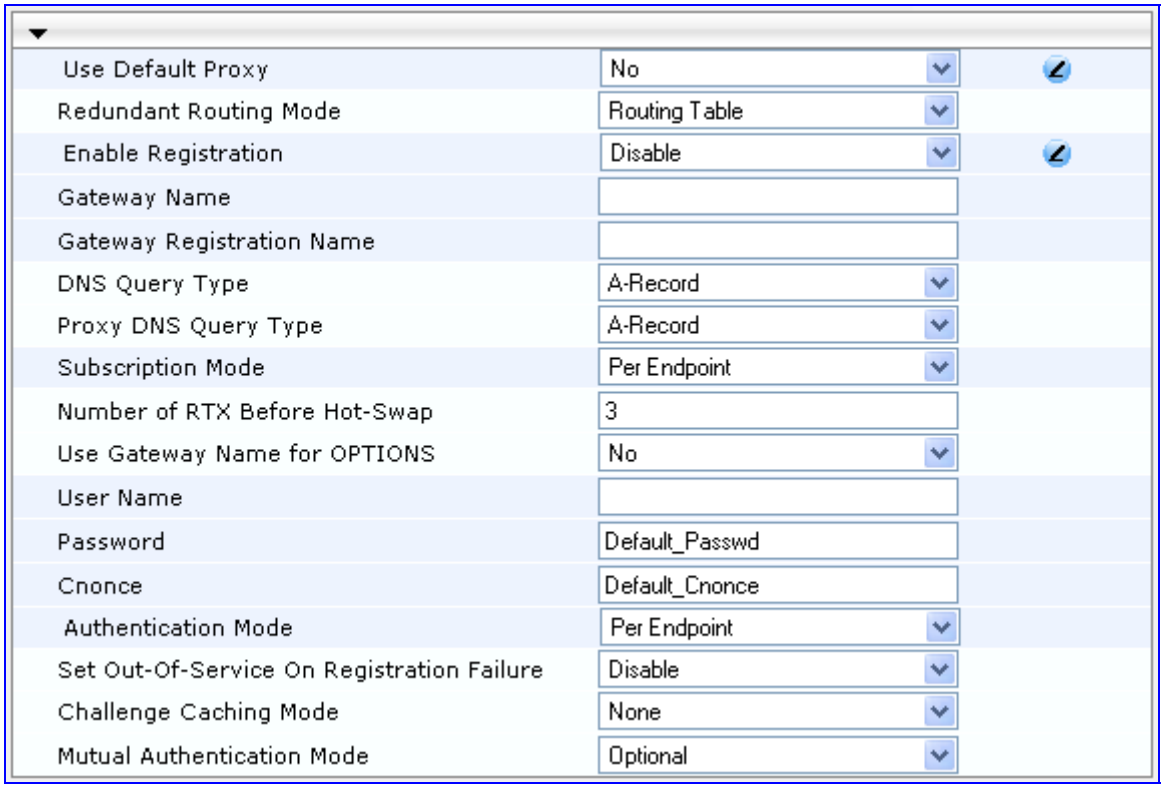
3.4.5.1.2 Proxy & Registration Parameters

The 'Proxy & Registration' page allows you to configure parameters that are associated with Proxy and Registration.

Note: To view whether the device or its endpoints have registered to a SIP Registrar/Proxy server, refer to 'Registration Status" on page 287.

➢ **To configure the Proxy & Registration parameters, take these 4 steps:**

1. Open the 'Proxy & Registration' page (**Configuration** tab > **Protocol Configuration** menu > **Protocol Definition** submenu > **Proxy & Registration** page item).



2. Configure the Proxy and Registration parameters according to the following table.
3. Click the **Submit** button to save your changes, or click the **Register** or **Un-Register** buttons to save your changes and register / unregister to a Proxy / Registrar.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-32: Proxy & Registration Parameters

Parameter	Description
Use Default Proxy **[IsProxyUsed]**	Enables the use of a SIP Proxy server. ▪ **[0]** Don't Use Proxy = Proxy isn't used - the internal routing table is used instead (default). ▪ **[1]** Use Proxy = Proxy is used. Parameters relevant to Proxy configuration are displayed. If you are using a Proxy server, enter the IP address of the Proxy server in the 'Proxy Sets table' (refer to "Proxy Sets Table" on page 153). If you are not using a Proxy server, you must configure the device's 'Tel to IP Routing' table (described in "Tel to IP Routing Table" on page 200).
Proxy Parameters (**Note:** These parameters appear only if the 'Use Default Proxy' parameter is enabled.)	
Proxy Set Table	Click the right-pointing arrow button to open the 'Proxy Sets Table' page to configure groups of proxy addresses. Alternatively, you can open this page from the **Proxy Sets Table** page item (refer to "Proxy Sets Table" on page 153 for a description of this page).
Proxy Name **[ProxyName]**	Defines the Home Proxy Domain Name. If specified, the Proxy Name is used as the Request-URI in REGISTER, INVITE, and other SIP messages, and as the host part of the To header in INVITE messages. If not specified, the Proxy IP address is used instead. The value must be string of up to 49 characters.
Redundancy Mode **[ProxyRedundancyMode]**	Determines whether the device switches back to the primary Proxy after using a redundant Proxy. ▪ **[0]** Parking = device continues working with a redundant (now active) Proxy until the next failure, after which it works with the next redundant Proxy (default). ▪ **[1]** Homing = device always tries to work with the primary Proxy server (i.e., switches back to the primary Proxy whenever it's available). **Note:** To use ProxyRedundancyMode, enable Keep-alive with Proxy option (EnableProxyKeepAlive = 1 or 2).
Proxy IP List Refresh Time **[ProxyIPListRefreshTime]**	Defines the time interval (in seconds) between each Proxy IP list refresh. The range is 5 to 2,000,000. The default interval is 60.
Enable Fallback to Routing Table **[IsFallbackUsed]**	Determines whether the device falls back to the 'Tel to IP Routing' table when Proxy servers are unavailable. ▪ **[0]** Disable = Fallback is not used (default). ▪ **[1]** Enable = 'Tel to IP Routing' table is used when Proxy servers are unavailable. When the device falls back to the internal 'Tel to IP Routing' table, the device continues scanning for a Proxy. When the device locates an active Proxy, it switches from internal routing back to Proxy routing. **Note:** To enable the redundant Proxies mechanism, set the parameter EnableProxyKeepAlive to 1 or 2.

Parameter	Description
Prefer Routing Table **[PreferRouteTable]**	Determines if the 'Tel to IP Routing' table takes precedence over a Proxy for routing calls. ▪ **[0]** No = Only Proxy is used to route calls (default). ▪ **[1]** Yes = The device checks the 'Dest Phone Prefix' and/or 'Source Phone Prefix' fields in the 'Tel to IP Routing' table for a match with the Tel-to-IP call. Only if a match is not found, a Proxy is used. **Note:** Applicable only if Proxy is not always used (AlwaysSendToProxy = 0, SendInviteToProxy = 0).
Use Routing Table for Host Names and Profiles **[AlwaysUseRouteTable]**	Use the internal 'Tel to IP Routing' table to obtain the URI Host name and (optionally) an IP profile (per call), even if a Proxy server is used. ▪ **[0]** Disable = Don't use (default). ▪ **[1]** Enable = Use the internal 'Tel to IP Routing' table. **Note:** This domain name is used instead of a Proxy name or IP address in the INVITE SIP URI.
Always Use Proxy **[AlwaysSendToProxy]**	Determines whether the device sends SIP messages and responses through a Proxy server. ▪ **[0]** Disable = Use standard SIP routing rules (default). ▪ **[1]** Enable = All SIP messages and responses are sent to a Proxy server. **Note:** Applicable only if Proxy server is used (i.e., the parameter IsProxyUsed is set to 1).
Send All INVITE to Proxy **[SendInviteToProxy]**	Determines whether INVITE messages, as a result of Transfer or Redirect are sent to a Proxy server instead of directly to the URI. ▪ **[0]** No = (default) INVITE messages from Transfer or Redirect are sent directly to the URI (according to the Refer-To header in the REFER message or Contact header in 30x response). ▪ **[1]** Yes = All INVITE messages, including those generated as a result of Transfer or Redirect are sent to a Proxy server. **Note:** Applicable only if a Proxy server is used and the parameter AlwaysSendtoProxy is set to 0.
Proxy / Registrar Registration parameters (**Note:** The proxy and registrar parameter fields appear only if 'Enable Registration' is enabled.)	
Enable Registration **[IsRegisterNeeded]**	Enables the device to register to a Proxy / Registrar server. ▪ **[0]** Disable = device doesn't register to Proxy / Registrar (default) server. ▪ **[1]** Enable = device registers to Proxy / Registrar server when the device is powered up and at every user-defined interval (configured by the parameter RegistrationTime). **Note:** The device sends a REGISTER request for each channel or for the entire device (according to the AuthenticationMode parameter).
Registrar Name **[RegistrarName]**	Registrar domain name. If specified, the name is used as the Request-URI in REGISTER messages. If it isn't specified (default), the Registrar IP address, or Proxy name or IP address is used instead. The valid range is up to 49 characters.

Parameter	Description
Registrar Transport Type **[RegistrarTransportType]**	Determines the transport layer used for outgoing SIP dialogs initiated by the device to the Registrar. ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When set to 'Not Configured', the value of the parameter SIPTransportType is used.
Registrar IP Address **[RegistrarIP]**	The IP address (or FQDN) and optionally, port number of the SIP Registrar server. The IP address is in dotted-decimal notation, e.g., 201.10.8.1:<5080>. **Notes:** ▪ If not specified, the REGISTER request is sent to the primary Proxy server. ▪ When a port number is specified, DNS NAPTR/SRV queries aren't performed, even if DNSQueryType is set to 1 or 2. ▪ If the RegistrarIP is set to an FQDN and is resolved to multiple addresses, the device also provides real-time switching (hotswap mode) between different Registrar IP addresses (IsProxyHotSwap is set to 1). If the first Registrar doesn't respond to the REGISTER message, the same REGISTER message is sent immediately to the next Proxy. EnableProxyKeepAlive must be set to 0 for this logic to apply. ▪ When a specific Transport Type is defined using RegistrarTransportType, a DNS NAPTR query is not performed even if DNSQueryType is set to 2.
Registration Time **[RegistrationTime]**	Defines the time interval (in seconds) for registering to a Proxy server. The value is used in the Expires header. In addition, this parameter defines the time interval between Keep-Alive messages when the parameter EnableProxyKeepAlive is set to 2 (REGISTER). Typically, the device registers every 3,600 sec (i.e., one hour). The device resumes registration according to the parameter RegistrationTimeDivider. The valid range is 10 to 2,000,000. The default value is 180.
Re-registration Timing [%] **[RegistrationTimeDivider]**	Defines the re-registration timing (in percentage). The timing is a percentage of the re-register timing set by the Registrar server. The valid range is 50 to 100. The default value is 50. For example: If RegistrationTimeDivider is 70% and Registration Expires time is 3600, the device re-sends its registration request after 3600 x 70% = 2520 sec. **Note**: This parameter may be overriden if the parameter RegistrationTimeThreshold is greater than 0 (refer to the description of RegistrationTimeThreshold).
Registration Retry Time **[RegistrationRetryTime]**	Defines the time interval (in seconds) after which a Registration request is resent if registration fails with a 4xx response or if there is no response from the Proxy/Registrar server. The default is 30 seconds. The range is 10 to 3600.

Parameter	Description
Registration Time Threshold **[RegistrationTimeThreshold]**	Defines a threshold (in seconds) for re-registration timing. If this parameter is greater than 0, but lower than the computed re-registration timing (according to the parameter RegistrationTimeDivider), the re-registration timing is set to the following: timing set by the Registration server in the Expires header minus the value of the parameter RegistrationTimeThreshold. The valid range is 0 to 2,000,000. The default value is 0.
Re-register On INVITE Failure **[RegisterOnInviteFailure]**	Enables immediate re-registration if a failure response is received for an INVITE request sent by the device. ▪ **[0]** Disable = Disabled (default) ▪ **[1]** Enable = Enabled
Miscellaneous parameters	
Gateway Name **[SIPGatewayName]**	Assigns a name to the device (e.g., 'gateway1.com'). Ensure that the name you choose is the one with which the Proxy is configured to identify the device. **Note:** If specified, the device name is used as the host part of the SIP URI in the From header. If not specified, the device's IP address is used instead (default).
Gateway Registration Name **[GWRegistrationName]**	Defines the user name that is used in the From and To headers in REGISTER messages. If no value is specified (default) for this parameter, the UserName parameter is used instead. **Note:** This parameter is applicable only for single registration per device (i.e., AuthenticationMode is set to 1). When the device registers each channel separately (i.e., AuthenticationMode is set to 0), the user name is set to the channel's phone number.
DNS Query Type **[DNSQueryType]**	Enables the use of DNS Naming Authority Pointer (NAPTR) and Service Record (SRV) queries to resolve Proxy and Registrar servers and to resolve all domain names that appear in the Contact and Record-Route headers. ▪ **[0]** A-Record = A-Record (default) ▪ **[1]** SRV = SRV ▪ **[2]** NAPTR = NAPTR If set to A-Record [0], no NAPTR or SRV queries are performed. If set to SRV [1] and the Proxy / Registrar IP address parameter, Contact / Record-Route headers, or IP address defined in the Routing tables contains a domain name, an SRV query is performed. The device uses the first host name received from the SRV query. The device then performs a DNS A-record query for the host name to locate an IP address. If set to NAPTR [2], an NAPTR query is performed. If it is successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is performed according to the configured transport type. If the Proxy / Registrar IP address parameter, the domain name in the Contact / Record-Route headers, or the IP address defined in the Routing tables contains a domain name with port definition, the device performs a regular DNS A-record query. If a specific Transport Type is defined, a NAPTR query is not performed. **Note:** To enable NAPTR/SRV queries for Proxy servers only, use the parameter ProxyDNSQueryType.

Parameter	Description
Proxy DNS Query Type **[ProxyDNSQueryType]**	Enables the use of DNS Naming Authority Pointer (NAPTR) and Service Record (SRV) queries to discover Proxy servers. ▪ **[0]** A-Record = A-Record (default) ▪ **[1]** SRV = SRV ▪ **[2]** NAPTR = NAPTR If set to A-Record [0], no NAPTR or SRV queries are performed. If set to SRV [1] and the Proxy IP address parameter contains a domain name without port definition (e.g., ProxyIP = domain.com), an SRV query is performed. The SRV query returns up to four Proxy host names and their weights. The device then performs DNS A-record queries for each Proxy host name (according to the received weights) to locate up to four Proxy IP addresses. Therefore, if the first SRV query returns two domain names, and the A-record queries return two IP addresses each, no additional searches are performed. If set to NAPTR [2], an NAPTR query is performed. If it is successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is performed according to the configured transport type. If the Proxy IP address parameter contains a domain name with port definition (e.g., ProxyIP = domain.com:5080), the device performs a regular DNS A-record query. If a specific Transport Type is defined, a NAPTR query is not performed. **Note:** When enabled, NAPTR/SRV queries are used to discover Proxy servers even if the parameter DNSQueryType is disabled.
Subscription Mode [SubscriptionMode]	Determines the method the device uses to subscribe to an MWI server. ▪ [0] Per Endpoint = Each endpoint subscribes separately - typically used for Mediant 1000 FXS interfaces (default). ▪ [1] Per Gateway = Single subscription for the entire device - typically used for Mediant 1000 FXO interfaces.
Use Gateway Name for OPTIONS **[UseGatewayNameForOptions]**	Determines whether the device uses its IP address or gateway name in keep-alive SIP OPTIONS messages. ▪ **[0]** No = Use the device's IP address in keep-alive OPTIONS messages (default). ▪ **[1]** Yes = Use 'Gateway Name' (SIPGatewayName) in keep-alive OPTIONS messages. The OPTIONS Request-URI host part contains either the device's IP address or a string defined by the parameter SIPGatewayName. The device uses the OPTIONS request as a keep-alive message to its primary and redundant Proxies (i.e., the parameter EnableProxyKeepAlive is set to 1).
Number of RTX Before Hot-Swap **[HotSwapRtx]**	Number of retransmitted INVITE/REGISTER messages before call is routed (hot swap) to another Proxy/Registrar. The valid range is 1 to 30. The default value is 3. **Note:** This parameter is also used for alternative routing using the 'Tel to IP Routing' table. If a domain name in the table is resolved into two IP addresses, and if there is no response for HotSwapRtx retransmissions to the INVITE message that is sent to the first IP address, the device immediately initiates a call to the second IP address.

Parameter	Description
User Name **[UserName]**	User name used for Registration and Basic/Digest authentication with a Proxy / Registrar server. The parameter doesn't have a default value (empty string). **Notes:** ▪ Applicable only if single device registration is used (i.e., Authentication Mode is set to Authentication Per gateway). ▪ For Mediant 1000: The Authentication table can be used instead (refer to Authentication on page 219).
Password **[Password]**	The password used for Basic/Digest authentication with a Proxy / Registrar server. A single password is used for all device ports. The default is 'Default_Passwd'. **Note:** For Mediant 1000: The Authentication table can be used instead (refer to Authentication on page 219).
Cnonce **[Cnonce]**	Cnonce string used by the SIP server and client to provide mutual authentication. (Free format, i.e., 'Cnonce = 0a4f113b'). The default is 'Default_Cnonce'.
Authentication Mode **[AuthenticationMode]**	Determines the device's registration and authentication method. ▪ **[0]** Per Endpoint = Registration and Authentication separately for each endpoint/B-channel. ▪ **[1]** Per Gateway = Single Registration and Authentication for the entire device (default). ▪ **[3]** Per FXS = Registration and Authentication for Mediant 1000 FXS endpoints. Typically, Authentication per endpoint is used for Mediant 1000 FXS interfaces,where each endpoint registers (and authenticates) separately with its own user name and password. Single Registration and Authentication (Authentication Mode = 1) is usually defined for FXO (for Mediant 1000) and digital modules.
Set Out-Of-Service On Registration Failure **[OOSOnRegistrationFail]**	Enables setting an endpoint or the entire device (i.e., all endpoints) to out-of-service if registration fails. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enabled. If the registration is per endpoint (i.e., AuthenticationMode is set to 0) and a specific endpoint registration fails (4xx or no response), then that endpoint is set to out-of-service until a success response is received in a subsequent registration request. When the registration is per the entire device (i.e., AuthenticationMode is set to 1) and registration fails, all endpoints are set to out-of-service. The out-of-service method is set according to the parameter FXSOOSBehavior. **Note:** This parameter is applicable only to Mediant 1000.

Parameter	Description
Challenge Caching Mode **[SIPChallengeCachingMode]**	Determines the mode for Challenge Caching, which reduces the number of SIP messages transmitted through the network. The first request to the Proxy is sent without authorization. The Proxy sends a 401/407 response with a challenge. This response is saved for further uses. A new request is resent with the appropriate credentials. Subsequent requests to the Proxy are automatically sent with credentials (calculated from the saved challenge). If the Proxy doesn't accept the new request and sends another challenge, the old challenge is replaced with the new one. ▪ **[0]** None = Challenges are not cached. Every new request is sent without preliminary authorization. If the request is challenged, a new request with authorization data is sent (default) ▪ **[1]** INVITE Only = Challenges issued for INVITE requests are cached. This prevents a mixture of REGISTER and INVITE authorizations. ▪ **[2]** Full = Caches all challenges from the proxies. **Note**: Challenge Caching is used with all proxies and not only with the active one.
Mutual Authentication Mode **[MutualAuthenticationMode]**	Determines the device's mode of operation when Authentication and Key Agreement (AKA) Digest Authentication is used. ▪ **[0]** Optional = Incoming requests that don't include AKA authentication information are accepted (default). ▪ **[1]** Mandatory = Incoming requests that don't include AKA authentication information are rejected.

3.4.5.1.3 Proxy Sets Table

The 'Proxy Sets Table' page allows you to define up to six groups of logical IP addresses (i.e., *Proxy Sets*), each containing up to five Proxy server IP addresses or FQDN's. In addition, Proxy load balancing and redundancy mechanisms can be applied per Proxy Set (if a Proxy Set contains more than one Proxy).

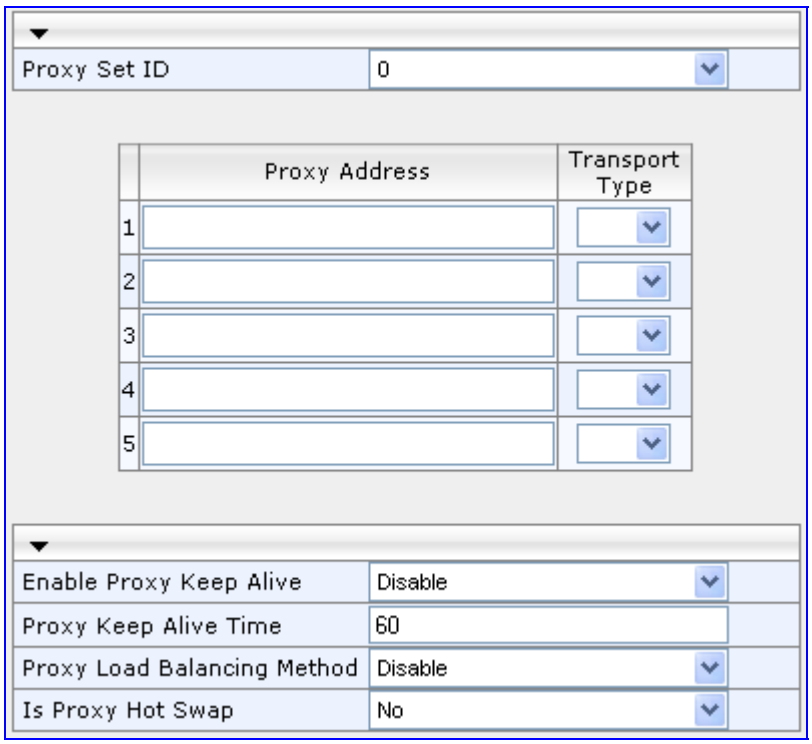
The Proxy Set ID can later be assigned to an IP Group (refer to "Configuring the IP Groups" on page 231). When the device sends an INVITE message to an IP Group, it is sent to the IP address of the Proxy Set associated with the specific IP Group.

Note: You can also configure the Proxy Sets table using the *ini* file table parameters ProxyIP and ProxySet (refer to "SIP Configuration Parameters" on page 323).

➢ To add Proxy servers and configure Proxy parameters, take these 5 steps:

1. Open the 'Proxy Sets Table' page (**Configuration** tab > **Protocol Configuration** menu > **Protocol Definition** submenu > **Proxy Sets Table** page item).

Figure 3-66: Proxy Sets Table Page



2. From the Proxy Set ID drop-down list, select an ID for the desired group.
3. Configure the Proxy parameters according to the following table.
4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-33: Proxy Sets Table Parameters

Parameter	Description
Proxy Set ID	The Proxy Set identification number. The valid range is 0 to 5 (i.e., up to 6 Proxy Set ID's can be configured). The Proxy Set ID #0 is used as the default Proxy Set, and if defined is backward compatible to the list of Proxies from earlier releases. You can define up to five IP addresses per Proxy Set. And per each IP address to define UDP, TCP or TLS transport type. **Note:** Although not recommended, you can use both default Proxy Set (ID #0) and IP Groups for call routing. For example, on the 'Trunk Group Settings' page (refer to "Configuring the Trunk Group Settings" on page 228), you can configure a Serving IP Group to where you want to route specific Trunk Group's channels, while all other device channels uses the default Proxy Set. At the same, you can also use IP Groups in the 'Tel to IP Routing' table (refer to "Tel to IP Routing Table" on page 200) to configure the default Proxy Set if the parameter PreferRouteTable is set to 1.

Parameter	Description
	To summarize, if the default Proxy Set is used, the INVITE message is sent according to the following preferences: ▪ To the Trunk Group's ServingIPGroupID, as defined in the 'Trunk Group Settings' table. ▪ According to the 'Tel to IP Routing' table, if the parameter PreferRouteTable is set to 1. ▪ To the default Proxy. Typically, when IP Groups are used, there is no need to use the default Proxy, and all routing and registration rules can be configured using IP Groups and the Account tables (refer to "Configuring the Account Table" on page 232).
Proxy Address	The IP address (and optionally port number) of the Proxy server. Up to five IP addresses can be configured per Proxy Set. Enter the IP address as an FQDN or in dotted-decimal notation (e.g., 201.10.8.1). You can also specify the selected port in the format: <IP Address>:<port>. If you enable Proxy Redundancy (by setting the parameter EnableProxyKeepAlive to 1 or 2), the device can operate with multiple Proxy servers. If there is no response from the first (*primary*) Proxy defined in the list, the device attempts to communicate with the other (*redundant*) Proxies in the list. When a redundant Proxy is located, the device either continues operating with it until the next failure occurs, or reverts to the primary Proxy (refer to the parameter ProxyRedundancyMode). If none of the Proxy servers respond, the device goes over the list again. The device also provides real-time switching (Hot-Swap mode) between the primary and redundant proxies (refer to the parameter IsProxyHotSwap). If the first Proxy doesn't respond to the INVITE message, the same INVITE message is immediately sent to the next Proxy in the list. The same logic applies to REGISTER messages (if RegistrarIP is not defined). **Notes:** ▪ If EnableProxyKeepAlive is set to 1 or 2, the device monitors the connection with the Proxies by using keep-alive messages (OPTIONS or REGISTER). ▪ To use Proxy Redundancy, you must specify one or more redundant Proxies. ▪ When a port number is specified (e.g., domain.com:5080), DNS NAPTR/SRV queries aren't performed, even if ProxyDNSQueryType is set to 1 or 2.
Transport Type	The transport type per Proxy server. ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS ▪ **[-1]** = Undefined **Note:** If no transport type is selected, the value of the global parameter SIPTransportType is used (refer to "SIP General Parameters" on page 134).

Parameter	Description
Proxy Load Balancing Method **[ProxyLoadBalancingMethod]**	Enables the Proxy Load Balancing mechanism per Proxy Set ID. ▪ **[0]** Disable = Load Balancing is disabled (default). ▪ **[1]** Round Robin = Round Robin. ▪ **[2]** Random Weights = Random Weights. When the Round Robin algorithm is used, a list of all possible Proxy IP addresses is compiled. This list includes all IP addresses per Proxy Set, after necessary DNS resolutions (including NAPTR and SRV, if configured). After this list is compiled, the Proxy Keep-Alive mechanism (according to parameters EnableProxyKeepAlive and ProxyKeepAliveTime) tags each entry as 'offline' or 'online'. Load balancing is only performed on Proxy servers that are tagged as 'online'. All outgoing messages are equally distributed across the list of IP addresses. REGISTER messages are also distributed unless a RegistrarIP is configured. The IP addresses list is refreshed according to ProxyIPListRefreshTime. If a change in the order of the entries in the list occurs, all load statistics are erased and balancing starts over again. When the Random Weights algorithm is used, the outgoing requests are not distributed equally among the Proxies. The weights are received from the DNS server by using SRV records. The device sends the requests in such a fashion that each Proxy receives a percentage of the requests according to its' assigned weight. A single FQDN should be configured as a Proxy IP address. The Random Weights Load Balancing is not used in the following scenarios: ▪ The Proxy Set includes more than one Proxy IP address. ▪ The only Proxy defined is an IP address and not an FQDN. ▪ SRV is not enabled (DNSQueryType). ▪ The SRV response includes several records with a different Priority value.
Enable Proxy Keep Alive **[EnableProxyKeepAlive]**	Determines whether Keep-Alive with the Proxy is enabled or disabled. This parameter is configured per Proxy Set. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Using OPTIONS = Enables Keep-Alive with Proxy using OPTIONS. ▪ **[2]** Using REGISTER = Enable Keep-Alive with Proxy using REGISTER. If set to 'Using OPTIONS', the SIP OPTIONS message is sent every user-defined interval, as configured by the parameter ProxyKeepAliveTime. If set to 'Using REGISTER', the SIP REGISTER message is sent every user-defined interval, as configured by the parameter RegistrationTime. Any response from the Proxy, either success (200 OK) or failure (4xx response) is considered as if the Proxy is communicating correctly. **Notes:** ▪ This parameter must be set to 'Using OPTIONS' when Proxy redundancy is used. ▪ When this parameter is set to 'Using REGISTER', the homing redundancy mode is disabled. ▪ When the active proxy doesn't respond to INVITE messages sent by the device, the proxy is tagged as 'offline'. The behavior is similar to a Keep-Alive (OPTIONS or REGISTER) failure.

Parameter	Description
Proxy Keep Alive Time **[ProxyKeepAliveTime]**	Defines the Proxy keep-alive time interval (in seconds) between Keep-Alive messages. This parameter is configured per Proxy Set. The valid range is 5 to 2,000,000. The default value is 60. **Note:** This parameter is applicable only if the parameter EnableProxyKeepAlive is set to 1 (OPTIONS). When the parameter EnableProxyKeepAlive is set to 2 (REGISTER), the time interval between Keep-Alive messages is determined by the parameter RegistrationTime.
Is Proxy Hot-Swap **[IsProxyHotSwap]**	Enables the Proxy Hot-Swap redundancy mode per Proxy Set. ▪ **[0]** No = Disabled (default). ▪ **[1]** Yes = Proxy Hot-Swap mode is enabled. If Proxy Hot-Swap is enabled, the SIP INVITE/REGISTER message is initially sent to the first Proxy/Registrar server. If there is no response from the first Proxy/Registrar server after a specific number of retransmissions (configured by the parameter HotSwapRtx), the INVITE/REGISTER message is resent to the next redundant Proxy/Registrar server.

3.4.5.1.4 Coders

The 'Coders' page allows you to configure up to five coders (and their attributes) for the device. The first coder in the list is the highest priority coder and is used by the device whenever possible. If the far-end device cannot use the first coder, the device attempts to use the next coder in the list, and so forth.

Notes:

- The device always uses the packetization time requested by the remote side for sending RTP packets.
- For an explanation on V.152 support (and implementation of T.38 and VBD coders), refer to "Supporting V.152 Implementation" on page 405.
- You can also configure the Coders table using the *ini* file table parameter CoderName (refer to "SIP Configuration Parameters" on page 323).

The coders supported by the device are listed in the table below:

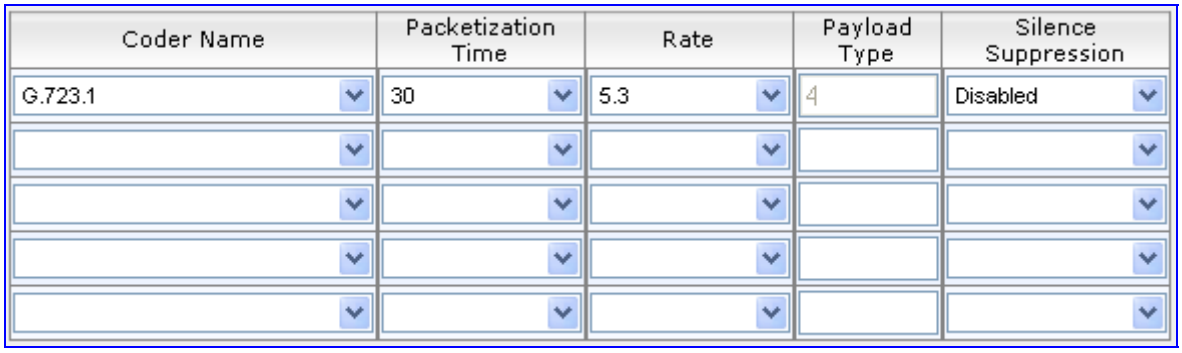
Table 3-34: Supported Coders

Coder Name	Packetization Time	Rate	Payload Type	Silence Suppression
G.711 A-law **[g711Alaw64k]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	Always 64	Always 8	• Disable **[0]** • Enable **[1]**
G.711 μ-law **[g711Ulaw64k]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	Always 64	Always 0	• Disable **[0]** • Enable **[1]**
EG.711 A-law **[eg711Alaw]**	10 (default), 20	Always 64	Dynamic (0-120)	N/A
EG.711 Mu-law **[eg711Ulaw]**	10 (default), 20	Always 64	Dynamic (0-120)	N/A
G.729 **[g729]**	10, 20 (default), 30, 40, 50, 60, 80, 100	Always 8	Always 18	• Disable **[0]** • Enable **[1]** • Enable w/o Adaptations **[2]**
G.723.1 **[g7231]**	30 (default), 60, 90, 120	5.3 **[0]**, 6.3 **[1]** (default)	Always 4	• Disable **[0]** • Enable **[1]**
G.726 **[g726]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	16 **[0]**, 24 **[1]**, 32 **[2]** (default) 40 **[3]**	Dynamic (0-120)	• Disable **[0]** • Enable **[1]**
iLBC **[iLBC]**	20 (default), 40, 60, 80, 100, 120	15 (default)	Dynamic (0-120)	• Disable **[0]** • Enable **[1]**
	30 (default), 60, 90, 120	13		
MS-GSM **[gsmMS]**	40 (default)	Always 13	Always 3	• Disable **[0]** • Enable **[1]**
NetCoder **[NetCoder]**	20 (default), 40, 60, 80, 100, 120	6.4 **[0]**; 7.2 **[1]** 8.0 **[2]** 8.8 **[3]** (default)	51 52 53 54	• Disable **[0]** • Enable **[1]**
G.711A-law_VBD **[g711AlawVbd]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	Always 64	Dynamic (0-120)	N/A
G.711U-law_VBD **[g711UlawVbd]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	Always 64	Dynamic (0-120)	N/A
T.38 **[t38fax]**	N/A	N/A	N/A	N/A

➢ To configure the device's coders, take these 9 steps:

1. Open the 'Coders' page (**Configuration** tab > **Protocol Configuration** menu > **Protocol Definition** submenu > **Coders** page item).

Figure 3-67: Coders Page



2. From the 'Coder Name' drop-down list, select the coder you want to use. For the full list of available coders and their corresponding attributes, refer to the table below.
3. From the 'Packetization Time' drop-down list, select the packetization time (in msec) for the selected coder. The packetization time determines how many coder payloads are combined into a single RTP packet.
4. From the 'Rate' drop-down list, select the bit rate (in kbps) for the selected coder.
5. In the 'Payload Type' field, if the payload type for the selected coder is dynamic, enter a value from 0 to 120 (payload types of 'well-known' coders cannot be modified). The payload type identifies the format of the RTP payload.
6. From the 'Silence Suppression' drop-down list, enable or disable the silence suppression option for the selected coder.
7. Repeat steps 2 through 6 for the second to fifth optional coders.
8. Click the **Submit** button to save your changes.
9. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Notes:

- Each coder (i.e., 'Coder Name') can appear only once.
- If packetization time and / or rate are not specified, the default value is applied.
- Only the packetization time of the first coder in the coder list is declared in INVITE / 200 OK SDP, even if multiple coders are defined.
- For G.729, it's also possible to select silence suppression without adaptations.
- If the coder G.729 is selected and silence suppression is disabled (for this coder), the device includes the string 'annexb=no' in the SDP of the relevant SIP messages. If silence suppression is enabled or set to 'Enable w/o Adaptations', 'annexb=yes' is included. An exception to this logic is when the remote gateway is a Cisco device (IsCiscoSCEMode).

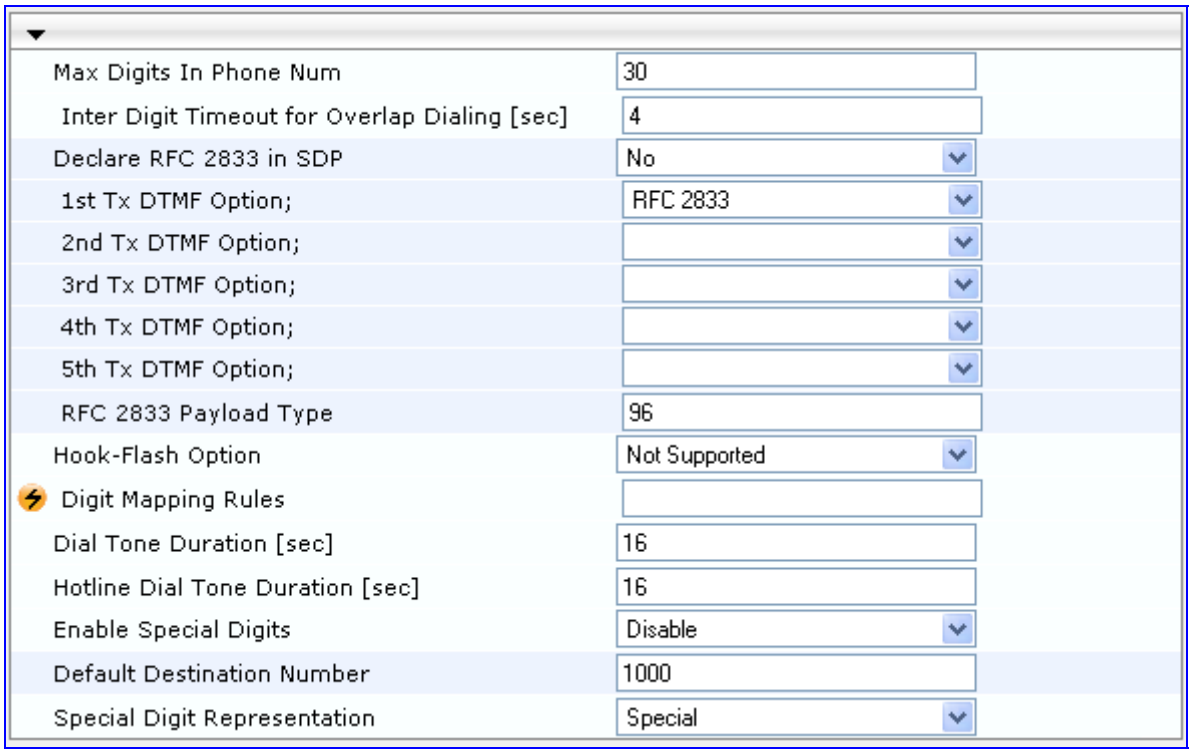
3.4.5.1.5 DTMF & Dialing Parameters

The 'DTMF & Dialing' page is used to configure parameters associated with dual-tone multi-frequency (DTMF) and dialing.

➢ **To configure the DTMF and dialing parameters, take these 4 steps:**

1. Open the 'DTMF & Dialing' page (**Configuration** tab > **Protocol Configuration** menu > **Protocol Definition** submenu > **DTMF & Dialing** page item).

Figure 3-68: DTMF & Dialing Page



2. Configure the DTMF and dialing parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-35: DTMF and Dialing Parameters

Parameter	Description
Max Digits in Phone Num **[MaxDigits]**	Defines the maximum number of collected destination number digits that can be received (i.e., dialed) from the Tel side when Tel-to-IP overlap dialing is performed (ISDN uses overlap dialing). When the number of collected digits reaches the maximum, the device uses these digits for the called destination number. The valid range is 1 to 49. The default value is 5 for Mediant 1000 analog and 30 for digital. **Notes:** ▪ Digit Mapping Rules can be used instead. ▪ For Mediant 1000: Dialing ends when the maximum number of digits is dialed, the Interdigit Timeout expires, the '#' key is dialed, or a digit map pattern is matched.
Inter Digit Timeout for Overlap Dialing [sec] **[TimeBetweenDigits]**	Defines the time (in seconds) that the device waits between digits that are received (i.e., dialed) from the Tel side when Tel-to-IP overlap dialing is performed (ISDN uses overlap dialing). When this inter-digit timeout expires, the device uses the collected digits for the called destination number. The valid range is 1 to 10. The default value is 4.
Declare RFC 2833 in SDP **[RxDTMFOption]**	Defines the supported Receive DTMF negotiation method. ▪ **[0]** No = Don't declare RFC 2833 telephony-event parameter in SDP. ▪ **[3]** Yes = Declare RFC 2833 telephony-event parameter in SDP (default). The device is designed to always be receptive to RFC 2833 DTMF relay packets. Therefore, it is always correct to include the 'telephony-event' parameter as default in the SDP. However, some devices use the absence of the 'telephony-event' in the SDP to decide to send DTMF digits in-band using G.711 coder. If this is the case, you can set RxDTMFOption to 0.

Parameter	Description
1st to 5th Tx DTMF Option **[TxDTMFOption]**	Determines a single or several preferred transmit DTMF negotiation methods. ▪ **[0]** Not Supported = No negotiation - DTMF digits are sent according to the parameters DTMFTransportType and RFC2833PayloadType (default). ▪ **[1]** INFO (Nortel) = Sends DTMF digits according to IETF <draft-choudhuri-sip-info-digit-00>. ▪ **[2]** NOTIFY = Sends DTMF digits according to <draft-mahy-sipping-signaled-digits-01>. ▪ **[3]** INFO (Cisco) = Sends DTMF digits according to Cisco format. ▪ **[4]** RFC 2833. ▪ **[5]** INFO (Korea) = Sends DTMF digits according to Korea Telecom format. **Notes:** ▪ DTMF negotiation methods are prioritized according to the order of their appearance. ▪ When out-of-band DTMF transfer is used ([1], [2], [3], or [5]), the parameter DTMFTransportType is automatically set to 0 (DTMF digits are erased from the RTP stream). ▪ When RFC 2833 (4) is selected, the device: 1) Negotiates RFC 2833 Payload Type (PT) using local and remote SDPs. 2) Sends DTMF packets using RFC 2833 PT according to the PT in the received SDP. 3) Expects to receive RFC 2833 packets with the same PT as configured by the parameter RFC2833PayloadType. 4) Sends DTMF digits in transparent mode (as part of the voice stream). ▪ When TxDTMFOption is set to 0, the RFC 2833 PT is set according to the parameter RFC2833PayloadType for both transmit and receive. ▪ The *ini* file table parameter TxDTMFOption can be repeated 5 times for configuring the DTMF transmit methods.
RFC 2833 Payload Type **[RFC2833PayloadType]**	The RFC 2833 DTMF relay dynamic payload type. The valid range is 96 to 99, and 106 to 127. The default is 96. The 100, 102 to 105 range is allocated for proprietary usage. **Notes:** ▪ Certain vendors (e.g., Cisco) use payload type 101 for RFC 2833. ▪ When RFC 2833 payload type (PT) negotiation is used (the parameter TxDTMFOption is set to 4), this payload type is used for the received DTMF packets. If negotiation isn't used, this payload type is used for receive and for transmit.

Parameter	Description
Hook-Flash Option **[HookFlashOption]**	Determines the supported hook-flash Transport Type (i.e., method by which hook-flash is sent and received). ▪ **[0]** Not Supported = Hook-Flash indication isn't sent (default). ▪ **[1]** INFO = Send proprietary INFO message with Hook-Flash indication. ▪ **[4]** RFC 2833 ▪ **[5]** INFO (Lucent) = Send proprietary INFO message with Hook-Flash indication. **Notes:** ▪ This parameter is applicable only to Mediant 1000. ▪ The RFC 2833 (4) option is currently not supported by digital interfaces. ▪ The DTMF HookFlashCode is send to IP according to the parameter HookFlashOption for digital interfaces. ▪ The FXO interfaces support the receipt of RFC 2833 Hook-Flash signals. ▪ The FXS interfaces send Hook-Flash signals only if EnableHold is set to 0.
Digit Mapping Rules **[DigitMapping]**	Digit map pattern (used to reduce the dialing period when Overlap dialing is used). If the digit string (i.e., dialed number) matches one of the patterns in the digit map, the device stops collecting digits and establishes a call with the collected number. The digit map pattern can contain up to 52 options, each separated by a vertical bar (|). The maximum length of the entire digit pattern is 152 characters. Available notations: • **[n-m]:** Range of numbers (not letters). • **.** (single dot)**:** Repeat digits until next notation (e.g., T). • **x:** Any single digit. • **T:** Dial timeout (configured by the parameter TimeBetweenDigits). • **S:** Immediately applies a specific rule that is part of a general rule. For example, if your digit map includes a general rule 'x.T' and a specific rule '11x', for the specific rule to take precedence over the general rule, append 'S' to the specific rule (i.e., '11xS'). An example of a digit map is shown below: 11xS|00T|[1-7]xxx|8xxxxxxx|#xxxxxxx|*xx|91xxxxxxxxx|9011x.T In the example above, the last rule can apply to International numbers - 9 for dialing tone, 011 Country Code, and then any number of digits for the local number ('x.'). **Note:** For BRI/PRI interfaces, the digitmap mechanism is applicable only when ISDN Overlap dialing is used (ISDNRxOverlap is set to 1).

Parameter	Description
Dial Tone Duration [sec] **[TimeForDialTone]**	Duration (in seconds) that the dial tone is played (for digital interface: to an ISDN terminal). For digital interfaces: This parameter is applicable for overlap dialing when ISDNInCallsBehavior = 65536. The dial tone is played if the ISDN SETUP message doesn't include the called number. The valid range is 0 to 60. The default is 5. For Mediant 1000 analog interfaces: FXS interface plays the dial tone after the phone is picked up (off-hook); while FXO interface plays the dial tone after port is seized in response to ringing (from PBX/PSTN). The default time is 16. **Notes for Mediant 1000 analog interfaces:** ▪ During play of dial tone, the device waits for DTMF digits. ▪ This parameter is not applicable when Automatic Dialing is enabled.
Hotline Dial Tone Duration **[HotLineToneDuration]**	Duration (in seconds) of the Hotline dial tone. If no digits are received during this duration, the device initiates a call to a user-defined number (refer to Automatic Dialing on page 220). The valid range is 0 to 60. The default is 16. **Note:** This parameter is applicable only to Mediant 1000 for both FXS and FXO interfaces.
Enable Special Digits **[IsSpecialDigits]**	Determines whether the asterisk (*) and pound (#) digits can be used. ▪ **[0]** Disable = Use '*' or '#' to terminate number collection (refer to the parameter UseDigitForSpecialDTMF). (Default.) ▪ **[1]** Enable = Allows '*' and '#' for telephone numbers dialed by a user or for the endpoint telephone number. **Notes:** ▪ This parameter is applicable only to Mediant 1000. ▪ These symbols can always be used as the first digit of a dialed number, even if you disable this parameter.
Default Destination Number **[DefaultNumber]**	Defines the default destination phone number used if the received message doesn't contain a called party number and no phone number is configured in the 'Trunk Group' table (refer to "Configuring the Trunk Group Table" on page 226). The parameter is used as a starting number for the list of channels comprising all trunk groups in the device. The default value is 1000.
Special Digit Representation **[UseDigitForSpecialDTMF]**	Defines the representation for 'special' digits ('*' and '#') that are used for out-of-band DTMF signaling (using SIP INFO/NOTIFY). ▪ **[0]** Special = Uses the strings '*' and '#' (default). ▪ **[1]** Numeric = Uses the numerical values 10 and 11.

3.4.5.2 Configuring the SIP Advanced Parameters

The **SIP Advanced Parameters** submenu allows you to configure advanced SIP control protocol parameters. This submenu contains the following page items:

- Advanced Parameters (refer to "General Parameters" on page 166)
- Supplementary Services (refer to "Supplementary Services" on page 176)
- Metering Tones (refer to "Metering Tones" on page 182)
- Charge Codes (refer to "Charge Codes Table" on page 183)
- Keypad Features (refer to "Keypad Features" on page 184)
- Stand-Alone Survivability (refer to "Stand-Alone Survivability" on page 188)

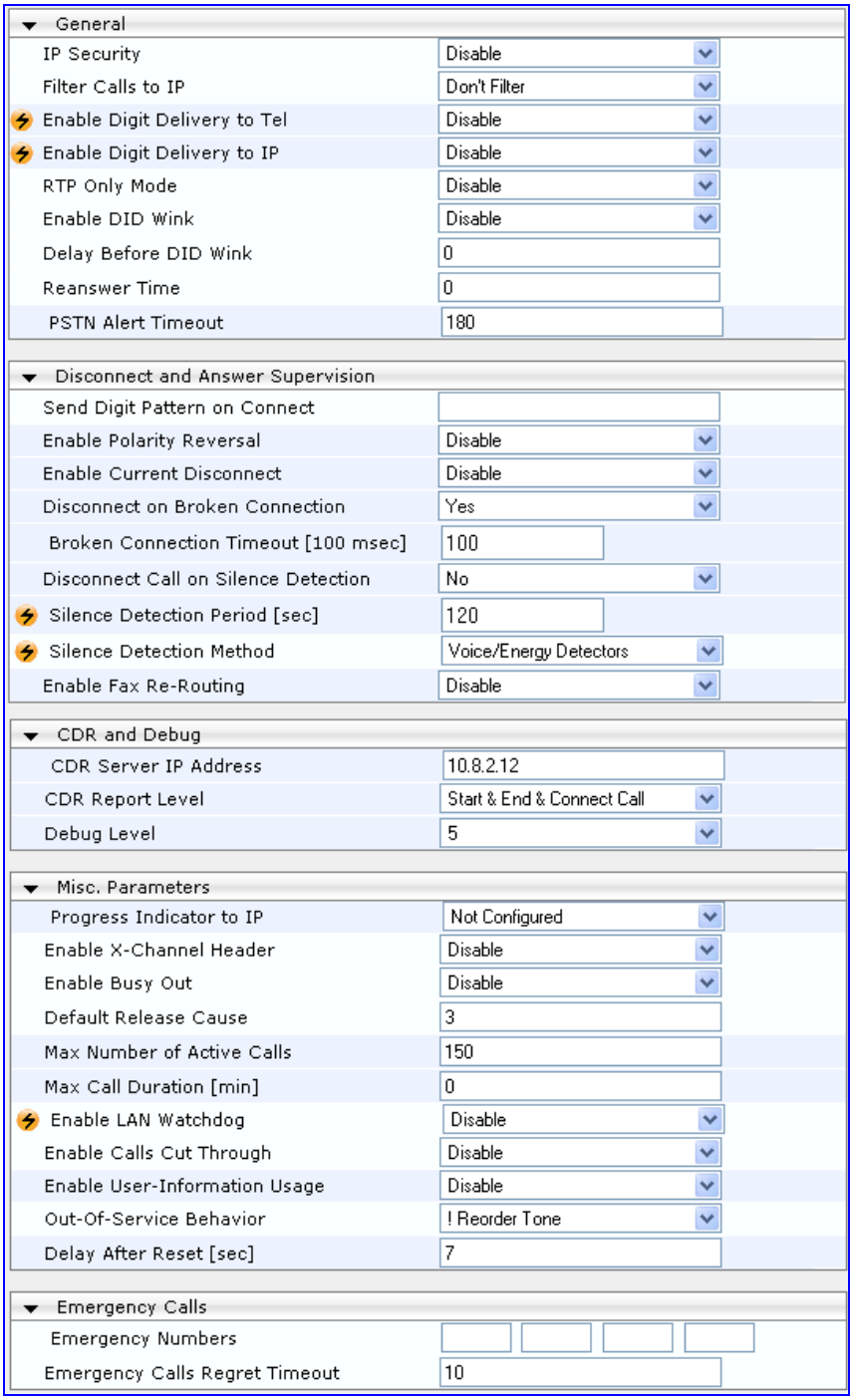
3.4.5.2.1 Advanced Parameters

The 'Advanced Parameters' page allows you to configure general control protocol parameters.

➢ **To configure the advanced general protocol parameters, take these 4 steps:**

1. Open the 'Advanced Parameters' page (**Configuration** tab > **Protocol Configuration** menu > **SIP Advanced Parameters** submenu > **Advanced Parameters** page item).

Figure 3-69: Advanced Parameters Page



2. Configure the parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-36: Advanced Parameters Description

Parameter	Description
General	
IP Security **[SecureCallsFromIP]**	Determines whether the device accepts SIP calls received only from IP addresses listed in the 'Tel to IP Routing' table (refer to "Tel to IP Routing Table" on page 200). This is useful in preventing unwanted SIP calls or messages and/or VoIP spam. ▪ **[0]** Disable = Device accepts all SIP calls (default). ▪ **[1]** Enable = Device accepts SIP calls only from IP addresses defined in the 'Tel to IP Routing' table. The device rejects all calls from unknown IP addresses. **Note:** Specifying the IP address of a Proxy server in the 'Tel to IP Routing' table enables the device to accept only calls originating from the Proxy server and reject all other calls that don't appear in this table.
Filter Calls to IP **[FilterCalls2IP]**	Enables filtering of Tel-to-IP calls when a Proxy is used (i.e., IsProxyUsed parameter is set to 1 -- refer to "Proxy & Registration Parameters" on page 146). ▪ **[0]** Don't Filter = device doesn't filter calls when using a Proxy. (default) ▪ **[1]** Filter = Filtering is enabled. When this parameter is enabled and a Proxy is used, the device first checks the 'Tel-to-IP Routing' table before making a call through the Proxy. If the number is not allowed (i.e., number isn't listed in the table or a call restriction routing rule of IP address 0.0.0.0 is applied), the call is released. **Note:** When no Proxy is used, this parameter must be disabled and filtering is according to the 'Tel-to-IP Routing' table.
Enable Digit Delivery to IP **[EnableDigitDelivery2IP]**	▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enable digit delivery to IP. The digit delivery feature enables sending DTMF digits to the destination IP address after the Tel-to-IP call is answered. To enable this feature, modify the called number to include at least one 'p' character.The device uses the digits before the 'p' character in the initial INVITE message. After the call is answered, the device waits for the required time (number of 'p' times 1.5 seconds) and then sends the rest of the DTMF digits using the method chosen (in-band or out-of-band). **Note:** The called number can include several 'p' characters (1.5 seconds pause), for example, 1001pp699, 8888p9p300.

Parameter	Description
Enable Digit Delivery to Tel **[EnableDigitDelivery]**	Enables the Digit Delivery feature, which sends DTMF digits (of the called number) to the device's port (phone line) after the call is answered [line offhooked (FXS) or seized (FXO)] for IP-to-Tel calls. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enable Digit Delivery feature for the FXO/FXS device. For digital modules: If the called number in IP-to-Tel call includes the characters 'w' or 'p', the device places a call with the first part of the called number (before 'w' or 'p') , and plays DTMF digits after the call is answered. If the character 'w' is used, the device waits for detection of dial tone before it starts playing DTMF digits. For example, if the called number is '1007766p100', the device places a call with 1007766 as the destination number, then after the call is answered, it waits 1.5 seconds ('p') and plays the rest of the number (100) as DTMF digits. Additional examples: 1664wpp102, 66644ppp503, and 7774w100pp200. **Notes for Mediant 1000 analog interfaces:** ▪ The called number can include characters 'p' (1.5 seconds pause) and 'd' (detection of dial tone). If character 'd' is used, it must be the first 'digit' in the called number. The character 'p' can be used several times. For example (for FXS/FXO interfaces), the called number can be as follows: d1005, dpp699, p9p300. To add the 'd' and 'p' digits, use the usual number manipulation rules. ▪ To use this feature with FXO interfaces, configure the device to operate in one-stage dialing mode. ▪ If this parameter is enabled, it is possible to configure the FXS/FXO interface to wait for dial tone per destination phone number (before or during dialing of destination phone number). Therefore, the parameter IsWaitForDialTone (configurable for the entire device) is ignored. ▪ The FXS interface send SIP 200 OK responses only after the DTMF dialing is complete.
RTP Only Mode **[RTPOnlyMode]**	Enables the device to start sending and/or receiving RTP packets to and from remote endpoints without the need to establish a Control session. The remote IP address is determined according to the 'Tel to IP Routing' table (refer to "Tel to IP Routing Table" on page 200). The port is the same port as the local RTP port (set by BaseUDPPort and the channel on which the call is received). ▪ **[0]** Disable = Disable (default). ▪ **[1]** Transmit & Receive = Send and receive RTP. ▪ **[2]** Transmit Only= Send RTP only. ▪ **[3]** Receive Only= Receive RTP only. **Notes:** ▪ To configure the RTP Only mode per trunk, use the RTPOnlyModeForTrunk_ID (refer to "Configuring the Trunk Settings" on page 96). ▪ If per trunk configuration (using RTPOnlyModeForTrunk) is set to other than default, the RTPOnlyMode parameter value is overridden.

Parameter	Description
PSTN Alert Timeout **[PSTNAlertTimeout]**	For Digital: Alert Timeout (in seconds) (ISDN T301 timer) for calls to PSTN. This timer is used between the time a SETUP message is sent to the Tel side (IP-to-Tel call establishment) and a CONNECT message is received. If an ALERTING message is received, the timer is restarted. For Mediant 1000 Analog: Alert Timeout (in seconds) for calls to the Tel side. This timer is used between the time ring is generated (FXS) or line is seized (FXO) until the call is connected. The default is 180 seconds. The range is 1 to 600. **Note:** If per trunk configuration (using TrunkPSTNAlertTimeout) is set to other than default (refer to "Configuring the Trunk Settings" on page 96), the PSTNAlertTimeout parameter value is overridden.
Enable DID Wink **[EnableDIDWink]**	Enables Direct Inward Dialing (DID) using Wink-Start signaling. ▪ **[0]** Disable = Disables DID Wink(default). ▪ **[1]** Enable = Enables DID Wink. If enabled, the device can be used for connection to EIA/TIA-464B DID Loop Start lines. Both FXO (detection) and FXS (generation) are supported. An FXO interface dials DTMF digits after a Wink signal is detected (instead of a Dial tone). An FXS interface generates the Wink signal after the detection of offhook (instead of playing a Dial tone). **Note:** This parameter is applicable only to Mediant 1000.
Delay Before DID Wink **[DelayBeforeDIDWink]**	Defines the time interval (in seconds) between detection of offhook and generation of a DID Wink. Applicable only to FXS interfaces. The valid range is 0 to 1,000. The default value is 0. **Note:** This parameter is applicable only to Mediant 1000.
Reanswer Time **[RegretTime]**	For Mediant 1000 Analog interfaces: The time interval after the user hangs up the phone and before the call is disconnected (FXS). This allows the user to hang up and then pick up the phone (before this timeout) to continue the call conversation. Thus, it's also referred to as regret time. For Digital interfaces: Determines the time period the device waits for an MFC R2 Resume (Reanswer) signal once a Suspend (Clear back) signal is received from the PBX. If this timer expires, the call is released. **Note:** Applicable only for MFC R2 CAS Brazil variant. The valid range is 0 to 255 (in seconds). The default value is 0.
Disconnect and Answer Supervision	
Send Digit Pattern on Connect **[TelConnectCode]**	Defines a digit pattern to send to the Tel side after SIP 200 OK is received from the IP side. The digit pattern is a pre-defined DTMF sequence that is used to indicate an answer signal (e.g., for billing). The valid range is 1 to 8 characters. **Note:** This parameter is applicable to FXO and CAS.

Parameter	Description
Enable Polarity Reversal **[EnableReversalPolarity]**	Enables the polarity reversal feature. ▪ **[0]** Disable = Disable the polarity reversal service (default). ▪ **[1]** Enable = Enable the polarity reversal service. If the polarity reversal service is enabled, the FXS interface changes the line polarity on call answer and then changes it back on call release. The FXO interface sends a 200 OK response when polarity reversal signal is detected (applicable only to one-stage dialing) and releases a call when a second polarity reversal signal is detected. **Note:** This parameter is applicable only to Mediant 1000.
Enable Current Disconnect **[EnableCurrentDisconnect]**	Enables call release upon detection of a current disconnect signal. ▪ **[0]** Disable = Disable the current disconnect service (default). ▪ **[1]** Enable = Enable the current disconnect service. If the current disconnect service is enabled, the FXO releases a call when a current disconnect signal is detected on its port, while the FXS interface generates a 'Current Disconnect Pulse' after a call is released from IP. The current disconnect duration is determined by the parameter CurrentDisconnectDuration. The current disconnect threshold (FXO only) is determined by the parameter CurrentDisconnectDefaultThreshold. The frequency at which the analog line voltage is sampled is determined by the parameter TimeToSampleAnalogLineVoltage. **Note:** This parameter is applicable only to Mediant 1000.
Disconnect on Broken Connection **[DisconnectOnBrokenConnection]**	Determines whether the device releases the call if RTP packets are not received within a user-defined timeout. ▪ **[0]** No ▪ **[1]** Yes (default) **Notes:** ▪ The timeout is set by the parameter BrokenConnectionEventTimeout. ▪ This feature is applicable only if the RTP session is used without Silence Compression. If Silence Compression is enabled, the device doesn't detect a broken RTP connection. ▪ During a call, if the source IP address (from where the RTP packets are sent) is changed without notifying the device, the device filters these RTP packets. To overcome this, set DisconnectOnBrokenConnection to 0; the device doesn't detect RTP packets arriving from the original source IP address and switches (after 300 msec) to the RTP packets arriving from the new source IP address.
Broken Connection Timeout **[BrokenConnectionEventTimeout]**	The time period (in 100 msec units) that an RTP packet is not received after which a call is disconnected. The valid range is 1 to 1,000. The default value is 100 (i.e., 10 seconds). **Notes:** ▪ Applicable only if DisconnectOnBrokenConnection = 1. ▪ Currently, this feature works only if Silence Suppression is disabled.

Parameter	Description
Disconnect Call on Silence Detection **[EnableSilenceDisconnect]**	Determines whether calls are disconnected after detection of silence. ▪ **[1]** Yes = The device disconnects calls in which silence occurs (in both call directions) for more than a user-defined time. ▪ **[0]** No = Call is not disconnected when silence is detected (default). The silence duration can be set by the FarEndDisconnectSilencePeriod parameter (default 120). **Note:** To activate this feature, set EnableSilenceCompression and FarEndDisconnectSilenceMethod to 1.
Silence Detection Period [sec] **[FarEndDisconnectSilencePeriod]**	Duration of silence period (in seconds) prior to call disconnection. The range is 10 to 28,800 (i.e., 8 hours). The default is 120 seconds.
Silence Detection Method **[FarEndDisconnectSilenceMethod]**	Silence detection method. ▪ **[0]** None = Silence detection option is disabled. ▪ **[1]** Packets Count = According to packet count. ▪ **[2]** Voice/Energy Detectors = N/A. ▪ **[3]** All = N/A.
Enable Fax Re-Routing **[EnableFaxReRouting]**	Enables or disables re-routing of Tel-to-IP calls that are identified as fax calls. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enabled. If a CNG tone is detected on the Tel side of a Tel-to-IP call, a 'FAX' prefix is appended to the destination number before routing and manipulations. An entry of 'FAX' as destination number in the 'Tel-to-IP Routing' table is then used to route the call, and the destination number manipulation mechanism is used to remove the 'FAX' prefix, if required. If the initial INVITE used to establish the voice call (not fax) was already sent, a CANCEL (if not connected yet) or a BYE (if already connected) is sent to tear down the voice call. **Notes:** ▪ To enable this feature, set CNGDetectorMode to 2, and IsFaxUsed to 1, 2, or 3. ▪ The 'FAX' prefix in routing and manipulation tables is case sensitive.
CDR and Debug	
CDR Server IP Address **[CDRSyslogServerIP]**	Defines the destination IP address to where CDR logs are sent. The default value is a null string, which causes CDR messages to be sent with all Syslog messages to the Syslog server. **Note:** The CDR messages are sent to UDP port 514 (default Syslog port).

Parameter	Description
CDR Report Level **[CDRReportLevel]**	Determines whether Call Detail Records (CDR) are sent to the Syslog server and when they are sent. ▪ **[0]** None = CDRs are not used (default). ▪ **[1]** End Call = CDR sent to the Syslog server at the end of each call. ▪ **[2]** Start & End Call = CDR report is sent to Syslog at the start and end of each call. ▪ **[3]** Connect & End Call = CDR report is sent to Syslog at connection and at the end of each call. ▪ **[4]** Start & Connect & End Call = CDR report is sent to Syslog at the start, at connection, and at the end of each call. The CDR Syslog message complies with RFC 3161 and is identified by: Facility = 17 (local1) and Severity = 6 (Informational).
Debug Level **[GwDebugLevel]**	Syslog debug logging level. ▪ **[0]** 0 = Debug is disabled (default). ▪ **[1]** 1 = Flow debugging is enabled. ▪ **[2]** 2 = Flow and device interface debugging are enabled. ▪ **[3]** 3 = Flow, device interface, and stack interface debugging are enabled. ▪ **[4]** 4 = Flow, device interface, stack interface, and session manager debugging are enabled. ▪ **[5]** 5 = Flow, device interface, stack interface, session manager, and device interface expanded debugging are enabled. **Note:** Usually set to 5 if debug traces are needed.
Misc. Parameters	
Progress Indicator to IP **[ProgressIndicator2IP]**	For Mediant 1000 Analog (FXS/FXO) interfaces: ▪ [0] No PI = For Tel-to-IP calls, the device sends 180 Ringing SIP response to IP after placing a call to a phone (FXS) or PBX (FXO). ▪ [1] PI = 1, [8] PI = 8: For Tel-to-IP calls, if EnableEarlyMedia = 1, the device sends 183 Session Progress message with SDP immediately after a call is placed to a phone/PBX. This is used to cut-through the voice path before the remote party answers the call, enabling the originating party to listen to network Call Progress Tones (such as ringback tone or other network announcements). ▪ [-1] Not Configured = Default values are used.The default for FXO interfaces is 1; The default for FXS interfaces is 0. For Digital (ISDN/CAS) interfaces: ▪ [-1] Not Configured = for ISDN spans, the progress indicator (PI) that is received in ISDN Proceeding, Progress, and Alert messages is used as described in the options below. (default) ▪ [0] No PI = For IP-to-Tel calls, the device sends 180 Ringing SIP response to IP after receiving ISDN Alert or (for CAS) after placing a call to PBX/PSTN. ▪ [1] PI =1, [8] PI =8: For IP-to-Tel calls, if EnableEarlyMedia = 1, the device sends 180 Ringing with SDP in response to an ISDN Alert or it sends a 183 Session Progress message with SDP in response to only the first received ISDN Proceeding or Progress message after a call is placed to PBX/PSTN over the trunk.

Parameter	Description
Enable X-Channel Header **[XChannelHeader]**	Determines whether the x-channel header is added to SIP messages for trunk / B-channel information. ▪ **[0]** Disable = x-channel header is not used (default). ▪ **[1]** Enable = x-channel header is generated with trunk/B-channel and IP address information. The header provides information on the E1/T1 physical trunk/B-channel on which the call is received or placed. For example, 'x-channel: DS/DS1-5/22;IP=192.168.13.1', where 'DS/DS-1' is a constant string, '5' is the trunk number, '22' is the B-channel, and in addition, the device's IP address is added to the header. This header is generated by the device and is sent in INVITE messages and 183/180/200OK responses.
Enable Busy Out **[EnableBusyOut]**	Determines whether the Busy Out feature is enabled. ▪ **[0]** Disable = 'Busy out' feature is not used (default). ▪ **[1]** Enable = 'Busy out' feature is enabled. When Busy Out is enabled and certain scenarios exist, the device performs the following: Mediant 1000 Analog interface: A reorder tone (determined by FXSOOSBehavior) is played when the phone is off-hooked. Digital interface: All E1/T1 trunks are automatically taken out of service by taking down the D-Channel or by sending a Service Out message for T1 PRI trunks supporting these messages (NI-2, 4/5-ESS, DMS-100, and Meridian). These behaviors are performed due to one of the following scenarios: ▪ Physically disconnected from the network (i.e., Ethernet cable is disconnected). ▪ The Ethernet cable is connected, but the device can't communicate with any host. Note that LAN Watch-Dog must be activated (EnableLANWatchDog = 1). ▪ The device can't communicate with the proxy (according to the Proxy keep-alive mechanism) and no other alternative exists to send the call. ▪ The IP Connectivity mechanism is enabled (using AltRoutingTel2IPEnable) and there is no connectivity to any destination IP address. **Notes for Mediant 1000 Analog interfaces:** ▪ The FXSOOSBehavior parameter controls the behavior of the FXS endpoints when a Busy Out or Graceful Lock occurs. ▪ FXO endpoints during Busy Out and Lock are inactive. ▪ Refer to the LifeLineType parameter for complementary optional behavior. **Note:** The Busy Out behavior varies between different protocol types (for Digtal interfaces).

Parameter	Description
Default Release Cause **[DefaultReleaseCause]**	Default Release Cause (to IP) for IP-to-Tel calls when the device initiates a call release and an explicit matching cause for this release isn't found. The default release cause is NO_ROUTE_TO_DESTINATION (3). Other common values include NO_CIRCUIT_AVAILABLE (34), DESTINATION_OUT_OF_ORDER (27), etc. **Notes:** ▪ The default release cause is described in the Q.931 notation and is translated to corresponding SIP 40x or 50x values. For example, 404 for 3, 503 for 34, and 502 for 27. ▪ When the Trunk is disconnected or is not synchronized, the internal cause is 305. This cause is mapped, by default, to SIP 503 Reply. ▪ For SIP-to-Q.931 and Q.931-to-SIP release cause mapping, refer to "Release Reason Mapping" on page 459.
Delay After Reset [sec] **[GWAppDelayTime]**	Defines the time interval (in seconds) that the device's operation is delayed after a reset. The valid range is 0 to 45. The default value is 7 seconds. **Note:** This feature helps to overcome connection problems caused by some LAN routers or IP configuration parameters' modifications by a DHCP server.
Max Number of Active Calls **[MaxActiveCalls]**	Defines the maximum number of simultaneous active calls supported by the device. If the maximum number of calls is reached, new calls are not established. The default value is the maximum available channels (no restriction on the maximum number of calls). The valid range is 1 to 120.
Max Call Duration (min) **[MaxCallDuration]**	Defines the maximum call duration (in minutes). If this time expires, both sides of the call are released (IP and Tel). The valid range is 0 to 35,791. The default is 0 (i.e., no limitation).
Enable LAN Watchdog **[EnableLanWatchDog]**	Determines whether the LAN Watch-Dog feature is enabled**.** ▪ **[0]** Disable = Disable LAN Watch-Dog (default). ▪ **[1]** Enable = Enable LAN Watch-Dog. When LAN Watch-Dog is enabled, the device's overall communication integrity is checked periodically. If no communication for about 3 minutes is detected, the device performs a self test. If the self test succeeds, the problem is logical link down (i.e., Ethernet cable disconnected on the switch side), and the Busy Out mechanism is activated if enabled (EnableBusyOut = 1). If the self test fails, the device restarts to overcome internal fatal communication error. **Note:** Enable LAN Watchdog is relevant only if the Ethernet connection is full duplex.
Enable Calls Cut Through **[CutThrough]**	Enables users to receive incoming IP calls while the port is in off-hook state. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enabled. If enabled, the FXS interface answers the call and 'cuts through' the voice channel if there is no other active call on the port, even if the port is in off-hook state. When the call is terminated (by the remote party), the device plays a reorder tone for a user-defined time (configured by the parameter TimeForReorderTone) and is then ready to answer the next incoming call without on-hooking the phone.

Parameter	Description
	The waiting call is automatically answered by the device when the current call is terminated (configured by setting the parameter EnableCallWaiting to 1). **Note:** This option is applicable only to Mediant 1000 FXS interfaces.
Enable User-Information Usage **[EnableUserInfoUsage]**	Enables or disables usage of the User Information loaded to the device in the User Information auxiliary file. (For a description on User Information, refer to "Loading Auxiliary Files" on page 266.) ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enabled.
Out-Of-Service Behavior **[FXSOOSBehavior]**	Determines the behavior of undefined FXS endpoints as well as all FXS endpoints when a Busy Out condition exists. ▪ **[0]** None = Normal operation. No response is provided to undefined endpoints. A dial tone is played to FXS endpoints when a Busy Out condition exists. ▪ **[1]** Reorder Tone = The device plays a reorder tone to the connected phone / PBX (default). ▪ **[2]** Polarity Reversal = The device reverses the polarity of the endpoint, marking it unusable (relevant, for example, to PBX DID lines). This option can't be configured on-the-fly. ▪ **[3]** Reorder Tone + Polarity Reversal = Same as 2 and 3 combined. This option can't be configured on-the-fly. ▪ **[4]** Current Disconnect = The device disconnects the current of the FXS endpoint. This option can't be configured on-the-fly. **Note:** This parameter is applicable only to Mediant 1000.
Emergency Calls	
Emergency Numbers **[EmergencyNumbers]**	Defines a list of numbers which are defined as 'emergency numbers'. When one of these numbers is dialed, the outgoing INVITE message includes the Priority and Resource-Priority headers. If the user sets the phone on-hook, the call is not disconnected, but instead a Hold Re-INVITE request is sent to the remote party. Only if the remote party disconnects the call (i.e., a BYE is received) or a timer expires (set by the parameter EmergencyRegretTimeout) is the call terminated. The list can include up to four different numbers, where each number can be up to four digits long. Example: EmergencyNumbers = ‘100’,’911’,’112’ **Note:** Applicable only to Mediant 1000 FXS interfaces.
Emergency Calls Reanswer Timeout **[EmergencyRegretTimeout]**	Determines the time (in minutes) that the device waits before tearing-down an emergency call (defined by the parameter EmergencyNumbers). Until this time expires, an emergency call can only be disconnected by the remote party [(typically, by a Public Safety Answering Point (PSAP)]. The valid range is 1 to 30. The default value is 10. **Note:** Applicable only to Mediant 1000 FXS interfaces.

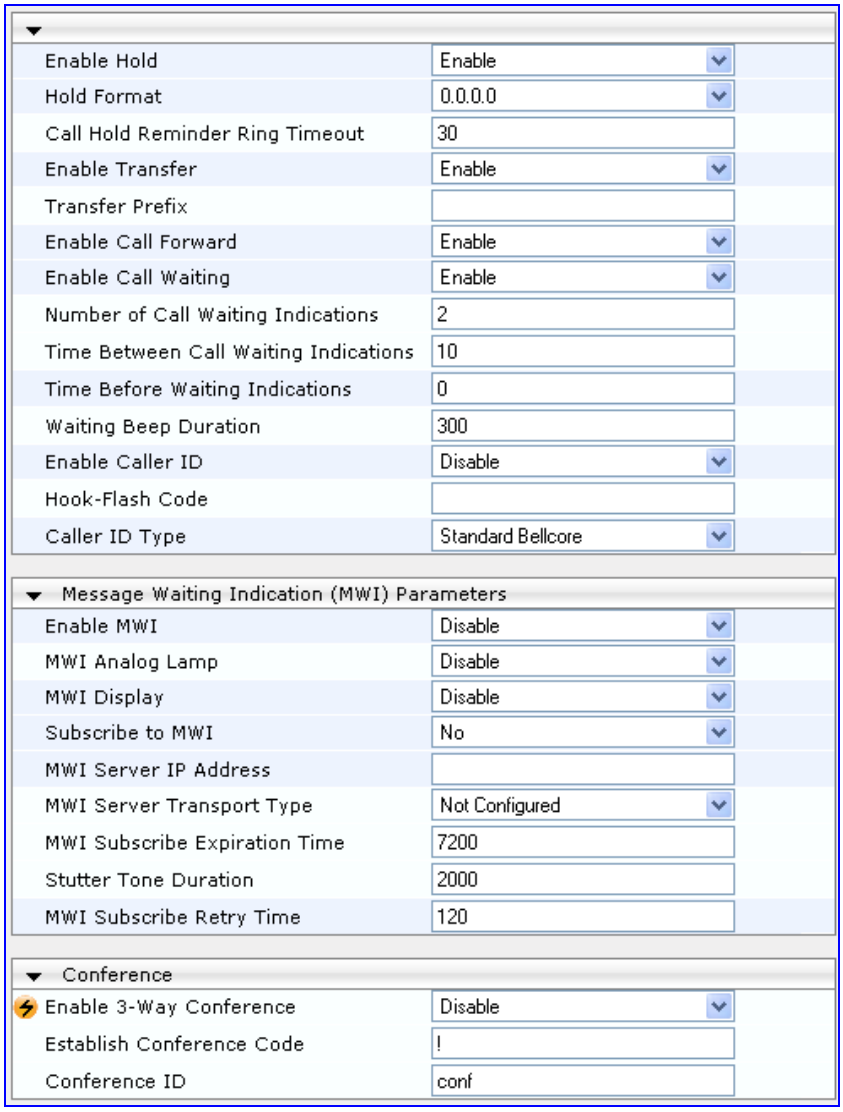
3.4.5.2.2 Supplementary Services

The 'Supplementary Services' page is used to configure parameters that are associated with supplementary services. For detailed information on supplementary services, refer to "Working with Supplementary Services" on page 437.

➢ **To configure the supplementary services' parameters, take these 4 steps:**

1. Open the 'Supplementary Services' page (**Configuration** tab > **Protocol Configuration** menu > **SIP Advanced Parameters** submenu > **Supplementary Services** page item).

Figure 3-70: Supplementary Services Page



2. Configure the supplementary services parameters according to the table below.
3. Click the **Submit** button to save your changes, or click the **Subscribe to MWI** or **Unsubscribe to MWI** buttons to save your changes and to subscribe / unsubscribe to the MWI server.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-37: Supplementary Services Parameters

Parameter	Description
Enable Hold **[EnableHold]**	Enables interworking of the Hold/Retrieve supplementary service from PRI to SIP. ▪ **[0]** Disable = Disables. ▪ **[1]** Enable = Enables (default). For Mediant 1000 analog: If the Hold service is enabled, a user can place the call on hold (or remove from hold) using the hook-flash. On receiving a Hold request, the remote party is placed on hold and hears the hold tone. **Notes:** ▪ This capability is only supported by the Euro ISDN variant and only from TE (user) to NT (network). ▪ To support interworking of the Hold/Retrieve supplementary service from SIP to ISDN, set EnableHold2ISDN to 1. ▪ To use this service, the Mediant analog devices at both ends must support this option.
Hold Format **[HoldFormat]**	Determines the format of the call hold request. ▪ **[0]** 0.0.0.0 = The connection IP address in SDP is 0.0.0.0 (default). ▪ **[1]** Send Only = The SDP contains the attribute 'a=sendonly'. **Note:** This parameter is applicable only to QSIG and Euro ISDN protocols.
Call Hold Reminder Ring Timeout **[CHRRTimeout]**	Defines the timeout (in seconds) for applying the Call Hold Reminder Ring. If a user hangs up while a call is still on hold, then the FXS interface immediately rings the extension for the duration specified by this parameter. If the user off-hooks the phone, the call becomes active. The valid range is 0 to 600. The default value is 30. **Note:** This parameter is applicable only to Mediant 1000 FXS interfaces.
Enable Transfer **[EnableTransfer]**	Determines whether call transfer is enabled. ▪ **[0]** Disable = Disable the call transfer service. ▪ **[1]** Enable = (default). If the transfer service is enabled, the user can activate Transfer using hook-flash signaling. If this service is enabled, the remote party performs the call transfer. (Applicable only to Mediant 1000 analog interfaces.) **Notes:** ▪ To use call transfer, the devices at both ends must support this option. ▪ To use call transfer, set the parameter EnableHold to 1.
Transfer Prefix **[xferPrefix]**	Defines the string that is added as a prefix to the transferred / forwarded called number when the REFER / 3xx message is received. **Notes:** ▪ The number manipulation rules apply to the user part of the REFER-TO / Contact URI before it is sent in the INVITE message. ▪ This parameter can be used to apply different manipulation rules to differentiate transferred / forwarded (only Mediant 1000 analog) number from the originally dialed number.

Parameter	Description
Enable Call Forward **[EnableForward]**	Determines whether Call Forward is enabled. ▪ **[0]** Disable = Disable the Call Forward service. ▪ **[1]** Enable = Enable Call Forward service (using REFER) (default). For Mediant 1000 FXS interfaces, the 'Call Forward' table (refer to "Call Forward" on page 222) must be defined to use the Call Forward service. **Note:** To use this service, the devices at both ends must support this option.
Enable Call Waiting **[EnableCallWaiting]**	Determines whether Call Waiting is enabled. ▪ **[0]** Disable = Disable the Call Waiting service. ▪ **[1]** Enable = Enable the Call Waiting service (default). If enabled, when an FXS interface (only Mediant 1000) receives a call on a busy endpoint, it responds with a 182 response (and not with a 486 busy). The device plays a call waiting indication signal. When hook-flash is detected, the device switches to the waiting call. The device that initiated the waiting call plays a Call Waiting Ringback tone to the calling party after a 182 response is received. **Notes:** ▪ The device's Call Progress Tones file must include a Call Waiting Ringback tone (caller side) and a Call Waiting tone (called side, FXS only). ▪ The EnableHold parameter must be enabled on both the calling and the called side. ▪ For Mediant 1000 analog: You can use the *ini* file table parameter CallWaitingPerPort to enable Call Waiting per port (refer to "Call Waiting" on page 225). ▪ For information on the Call Waiting feature, refer to "Call Waiting" on page 440. ▪ For information on the Call Progress Tones file, refer to Configuring the Call Progress Tones File.
Number of Call Waiting Indications **[NumberOfWaitingIndications]**	Number of Call Waiting indications that are played to the called telephone that is connected to the device (FXS only) for Call Waiting. The valid range is 1 to 100 indications. The default value is 2. **Note:** This parameter is applicable only to Mediant 1000.
Time Between Call Waiting Indications **[TimeBetweenWaitingIndications]**	Time (in seconds) between consecutive call waiting indications (FXS only) for call waiting. The valid range is 1 to 100. The default value is 10. **Note:** This parameter is applicable only to Mediant 1000.
Time Before Waiting Indications **[TimeBeforeWaitingIndication]**	Defines the interval (in seconds) before a call waiting indication is played to the port that is currently in a call (FXS only). The valid range is 0 to 100. The default time is 0 seconds. **Note:** This parameter is applicable only to Mediant 1000.
Waiting Beep Duration **[WaitingBeepDuration]**	Duration (in msec) of call waiting indications that are played to the port that is receiving the call (FXS only). The valid range is 100 to 65535. The default value is 300. **Note:** This parameter is applicable only to Mediant 1000.

Parameter	Description
Enable Caller ID **[EnableCallerID]**	Determines whether Caller ID is enabled. ▪ **[0]** Disable = Disable the Caller ID service (default). ▪ **[1]** Enable = Enable the Caller ID service. If the Caller ID service is enabled, then for FXS interfaces, calling number and Display text (from IP) are sent to the device's port. For FXO interfaces, the Caller ID signal is detected and sent to IP in the SIP INVITE message (as 'Display' element). For information on the Caller ID table, refer to "Caller ID" on page 221. To disable/enable caller ID generation per port, refer to "Call Forward" on page 222. **Note:** This parameter is applicable only to Mediant 1000.
Caller ID Type **[CallerIDType]**	Defines one of the following standards for detection (FXO) and generation (FXS) of Caller ID, and detection (FXO) generation (FXS) of MWI (when specified) signals: ▪ **[0]** Standard Bellcore = Caller ID and MWI (default) ▪ **[1]** Standard ETSI = Caller ID and MWI ▪ **[2]** Standard NTT ▪ **[4]** Standard BT = Britain ▪ **[16]** Standard DTMF Based ETSI ▪ **[17]** Standard Denmark = Caller ID and MWI ▪ **[18]** Standard India ▪ **[19]** Standard Brazil **Notes:** ▪ This parameter is applicable only to Mediant 1000. ▪ Typically, the Caller ID signals are generated/detected between the first and second rings. However, sometimes the Caller ID is detected before the first ring signal (in such a scenario, configure RingsBeforeCallerID to 0). ▪ Caller ID detection for Britain [4] is not supported on the device's FXO ports. Only FXS ports can generate the Britain [4] Caller ID. ▪ To select the Bellcore Caller ID sub standard, use the parameter BellcoreCallerIDTypeOneSubStandard. To select the ETSI Caller ID substandard, use the parameter ETSICallerIDTypeOneSubStandard. ▪ To select the Bellcore MWI sub standard, use the parameter BellcoreVMWITypeOneStandard. To select the ETSI MWI sub standard, use the parameter ETSIVMWITypeOneStandard.
Hook-Flash Code **[HookFlashCode]**	Determines the digit pattern used by the PBX to indicate a Hook Flash event. When this pattern is detected from the Tel side, the device responds as if a Hook Flash event occurs and sends a SIP INFO message if HookFlashOption is set to 1, indicating Hook Flash. If configured and a Hook Flash indication is received from the IP side, the device generates this pattern to the Tel side. The valid range is a 25-character string. The default is a null string.

Parameter	Description
Message Waiting Indication (MWI) Parameters **Note:** MWI is applicable only to Mediant 1000.	
Enable MWI **[EnableMWI]**	Enables Message Waiting Indication (MWI). ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = MWI service is enabled. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ The device supports only the receipt of SIP MWI NOTIFY messages (the device doesn't generate these messages). ▪ For detailed information on MWI, refer to "Message Waiting Indication" on page 440.
MWI Analog Lamp **[MWIAnalogLamp]**	Enables visual display of MWI. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Enable = Enables visual Message Waiting Indication by supplying line voltage of approximately 100 VDC to activate the phone's lamp. **Note:** This parameter is applicable only for FXS interfaces.
MWI Display **[MWIDisplay]**	Determines whether MWI information is sent to the phone display. ▪ **[0]** Disable = MWI information isn't sent to display (default). ▪ **[1]** Enable = The device generates an MWI message (determined by the parameter CallerIDType), which is displayed on the MWI display. **Note:** This parameter is applicable only to FXS interfaces.
Subscribe to MWI **[EnableMWISubscription]**	Enables subscription to an MWI server. ▪ **[0]** No = Disables MWI subscription (default). ▪ **[1]** Yes = Enables subscription to MWI (to MWIServerIP address). **Note:** Use the parameter SubscriptionMode (described in Proxy & Registration Parameters on page 146) to determine whether the device subscribes per endpoint or per the entire device.
MWI Server IP Address **[MWIServerIP]**	MWI server's IP address. If provided, the device subscribes to this IP address. The MWI server address can be configured as a numerical IP address or as a domain name. If not configured, the Proxy IP address is used instead.
MWI Server Transport Type **[MWIServerTransportType]**	Determines the transport layer used for outgoing SIP dialogs initiated by the device to the MWI Server. ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When set to 'Not Configured', the value of the parameter SIPTransportType is used.
MWI Subscribe Expiration Time **[MWIExpirationTime]**	The MWI subscription expiration time in seconds. The default is 7200 seconds. The range is 10 to 72000.

Parameter	Description
Stutter Tone Duration **[StutterToneDuration]**	Duration (in msec) of the played stutter dial tone, indicating enabled Call Forward or waiting message(s). The Stutter tone is played (instead of a regular dial tone), when Call Forward is enabled on the specific port or when Message Waiting Indication (MWI) is received. The tone is composed of a Confirmation tone, which is played for a user-defined duration (StutterToneDuration), followed by a Stutter tone. Both tones are defined in the CPT file. After this duration, a dial tone is played. The range is 1,000 to 60,000. The default is 2,000 (i.e., 2 seconds). **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ The MWI tone takes precedence over the Call Forwarding Reminder tone. For detailed information on MWI, refer to "Message Waiting Indication" on page 440.
MWI Subscribe Retry Time **[SubscribeRetryTime]**	Subscription retry time (in seconds) after last subscription failure. The default is 120 seconds. The range is 10 to 7200.
Conference Parameters	
Enable 3-Way Conference **[Enable3WayConference]**	Enables or disables the 3-Way Conference feature. ▪ **[0]** Disable = Disable (default) ▪ **[1]** Enable = Enables 3-way conferencing **Note:** This parameter is applicable only to Mediant 1000.
Establish Conference Code **[ConferenceCode]**	Defines the digit pattern, which upon detection generates the Conference-initiating INVITE when 3-way conferencing is enabled (Enable3WayConference is set to 1). The valid range is a 25-character string. The default is “!” (Hook-Flash). **Note:** This parameter is applicable only to Mediant 1000.
Conference ID **[ConferenceID]**	Defines the Conference Identification string (up to 16 characters). The default value is 'conf'. For 3-way conferencing using an external media server: The device uses this identifier in the Conference-initiating INVITE that is sent to the media server when Enable3WayConference is set to 1. When using the Mediant 1000 Media Processing Module (MPM): To join a conference, the INVITE URI must include the Conference ID string, preceded by the number of the participants in the conference, and terminated by a unique number. For example: INVITE sip:4MyConference1234@10.1.10.10. INVITE messages with the same URI join the same conference. For example: ConferenceID = MyConference. **Note:** This parameter is applicable only to Mediant 1000.

3.4.5.2.3 Metering Tones

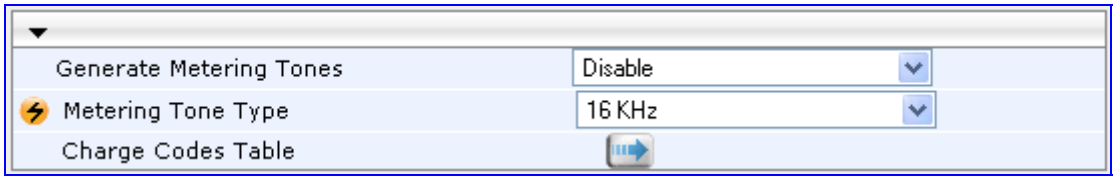
The FXS interfaces can generate 12/16 KHz metering pulses towards the Tel side (e.g., for connection to a payphone or private meter). Tariff pulse rate is determined according to an internal table. This capability enables users to define different tariffs according to the source / destination numbers and the time-of-day. The tariff rate includes the time interval between the generated pulses and the number of pulses generated on answer.

Note: The 'Metering Tones' page is only available if the device provides FXS interfaces (i.e., the FXS module installed) and is applicable only to Mediant 1000.

➢ **To configure the Metering tones, take these 4 steps:**

1. Open the 'Metering Tones' page (**Configuration** tab > **Protocol Configuration** menu > **SIP Advanced Parameters** submenu > **Metering Tones** page item).

Figure 3-71: Metering Tones Page



2. Configure the Metering tones parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to the flash memory, refer to "Saving Configuration" on page 265.

Note: In the 'Tel to IP Routing' table (refer to "Tel to IP Routing Table" on page 200), assign a charge code rule to the routing rules you require. When a new call is established, the 'Tel to IP Routing' table is searched for the destination IP addresses. Once a route is found, the Charge Code (configured for that route) is used to associate the route with an entry in the 'Charge Codes' table.

Table 3-38: Metering Tones Parameters

Parameter	Description
Generate Metering Tones **[PayPhoneMeteringMode]**	Determines the method used to configure the metering tones that are generated to the Tel side. ▪ **[0]** Disable = Metering tones aren't generated (default). ▪ **[1]** Internal Table = Metering tones are generated according to the internal table configured by the parameter ChargeCode. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ If you select 'Internal Table', you must configure the 'Charge Codes Table' (refer to "Charge Codes Table" on page 183).

Parameter	Description
Metering Tone Type **[MeteringType]**	Defines the metering tone (12 or 16 kHz) that is generated by FXS interfaces. ▪ **[0]** 12 kHz = 12-kHz metering tone (default). ▪ **[1]** 16 kHz = 16-kHz metering tone. **Note:** A suitable (12 or 16 KHz) *FXS Coefficient* file must be used for FXS interfaces.
Charge Codes Table	If you configured the 'Generate Metering Tones' parameter to 'Internal Table', access the 'Charge Codes Table' page, by clicking . For detailed information on configuring the Charge Codes table, refer to "Charge Codes Table" on page 183.

3.4.5.2.4 Charge Codes Table

The 'Charge Codes Table' page is used to configure the metering tones (and their time interval) that the FXS interfaces generate to the Tel side. To associate a charge code to an outgoing Tel-to-IP call, use the 'Tel to IP Routing' table.

Notes:

- The 'Charge Codes Table' page is only available if the device provides FXS interface (i.e., FXS module installed) and is applicable only to Mediant 1000.
- You can also configure the Charge Codes table using the *ini* file table parameter ChargeCode (refer to "Analog Telephony Parameters" on page 355).

➢ **To configure the Charge Codes table, take these 4 steps:**

1. Access the 'Charge Codes Table' page (**Configuration** tab > **Protocol Configuration** menu > **SIP Advanced Parameters** submenu > **Charge Codes** page item). Alternatively, you can also access this page from the 'Metering Tones' page (refer to "Metering Tones" on page 182).

Figure 3-72: Charge Codes Table Page

Table Index: 0-4

Index	Time Period 1			Time Period 2			Time Period 3			Time Period 4		
	End Time	Pulse Interval	Pulses On Answer	End Time	Pulse Interval	Pulses On Answer	End Time	Pulse Interval	Pulses On Answer	End Time	Pulse Interval	Pulses On Answer
0	07	30	1	14	20	2	20	15	1	00	60	1
1	05	60	1	14	20	1	00	60	1			
2	00	60	1									
3												
4												

2. Use the table to define up to 25 different charge codes (each charge code is defined per row). Each code can include from a single and up to four different time periods in a day (24 hours). Each time period is composed of:
   - The end of the time period (in a 24 rounded-hour's format).
   - The time interval between pulses (in seconds).
   - The number of pulses sent on answer.

   The first time period always starts at midnight (00). It is mandatory that the last time period of each rule ends at midnight (00). This prevents undefined time frames in a day. The device selects the time period by comparing the device 's current time to the end time of each time period of the selected Charge Code. The device generates the Number of Pulses on Answer once the call is connected and from that point on, it generates a pulse each Pulse Interval. If a call starts at a certain time period and crosses to the next, the information of the next time period is used.
3. Click the **Submit** button to save your changes.
4. To save the changes to the flash memory, refer to "Saving Configuration" on page 265.

3.4.5.2.5 Keypad Features

The 'Keypad Features' page (applicable only to FXS interfaces) enables you to activate and deactivate the following features directly from the connected telephone's keypad:

- Call Forward (refer to "Call Forward" on page 222)
- Caller ID Restriction (refer to "Caller ID" on page 221)
- Hotline (refer to "Automatic Dialing" on page 220)

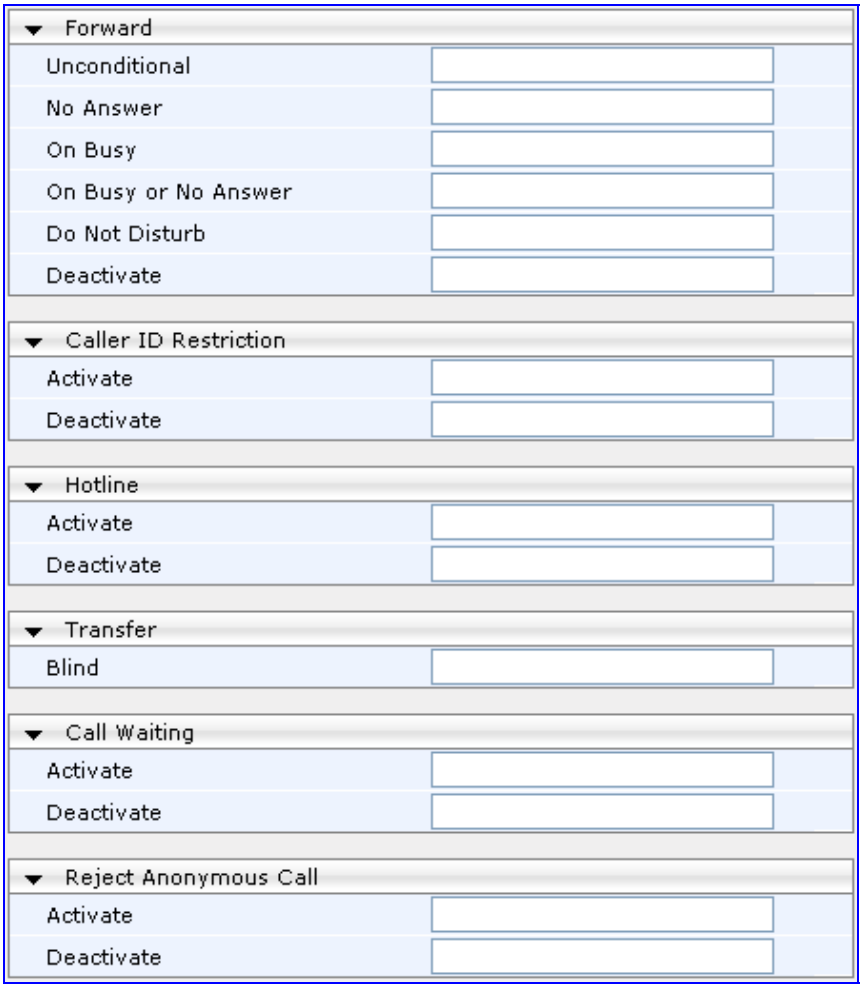
Notes:

- The 'Keypad Features' page is only available if the device supports FXS interface and is applicable only to Mediant 1000.
- The method used by the device to collect dialed numbers is identical to the method used during a regular call (i.e., max digits, interdigit timeout, digit map, etc.).
- The activation of each feature remains in effect until it is deactivated (i.e., not deactivated after a call).

➢ To configure the keypad features, take these 4 steps:

1. Open the 'Keypad Features' page (**Configuration** tab > **Protocol Configuration** menu > **SIP Advanced Parameters** submenu > **Keypad Features** page item).

Figure 3-73: Keypad Features Page



2. Configure the Keypad Features according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to the flash memory, refer to "Saving Configuration" on page 265.

Table 3-39: Keypad Features Parameters Description

Parameter	Description
Forward (**Note:** The forward type and number can be viewed in the 'Call Forward' table - refer to "Call Forward" on page 222.)	
Unconditional **[KeyCFUnCond]**	Keypad sequence that activates the immediate call forward option.
No Answer **[KeyCFNoAnswer]**	Keypad sequence that activates the forward on no answer option.

Parameter	Description
On Busy **[KeyCFBusy]**	Keypad sequence that activates the forward on busy option.
On Busy or No Answer **[KeyCFBusyOrNoAnswer]**	Keypad sequence that activates the forward on 'busy or no answer' option.
Do Not Disturb **[KeyCFDoNotDisturb]**	Keypad sequence that activates the Do Not Disturb option (immediately reject incoming calls).
To activate the required forward method from the telephone: 1. Dial the preconfigured sequence number on the keypad; a dial tone is heard. 2. Dial the telephone number to which the call is forwarded (terminate the number with #); a confirmation tone is heard.	
Deactivate **[KeyCFDeact]**	Keypad sequence that deactivates any of the call forward options. After the sequence is pressed, a confirmation tone is heard.
Caller ID Restriction (**Note:** The caller ID presentation can be viewed in the 'Caller Display Information' table - refer to "Caller ID" on page 221.)	
Activate **[KeyCLIR]**	Keypad sequence that activates the restricted Caller ID option. After the sequence is pressed, a confirmation tone is heard.
Deactivate **[KeyCLIRDeact]**	Keypad sequence that deactivates the restricted Caller ID option. After the sequence is pressed, a confirmation tone is heard.
Hotline (**Note:** The destination phone number and the auto dial status can be viewed in the 'Automatic Dialing' table - refer to "Automatic Dialing" on page 220.)	
Activate **[KeyHotLine]**	Keypad sequence that activates the delayed hotline option. To activate the delayed hotline option from the telephone, perform the following: ▪ Dial the preconfigured sequence number on the keypad; a dial tone is heard. ▪ Dial the telephone number to which the phone automatically dials after a configurable delay (terminate the number with #); a confirmation tone is heard.
Deactivate **[KeyHotLineDeact]**	Keypad sequence that deactivates the delayed hotline option. After the sequence is pressed, a confirmation tone is heard.

Parameter	Description
Transfer	
Blind **[KeyBlindTransfer]**	Keypad sequence that activates blind transfer for Tel-to-IP calls. There are two possible scenarios: ▪ Option 1: After this sequence is dialed, the current call is put on hold (using Re-INVITE), a dial tone is played to the phone, and then phone number collection starts. ▪ Option 2: A Hook-Flash is pressed, the current call is put on hold, a dial tone is played to the phone, and then digit collection starts. After this sequence is identified, the device continues the collection of the destination phone number. For both options, after the phone number is collected, it's sent to the transferee in a SIP REFER request (without a Replaces header). The call is then terminated and a confirmation tone is played to the phone. If the phone number collection fails due to a mismatch, a reorder tone is played to the phone. **Notes:** ▪ This parameter is applicable to FXO and FXS interfaces, but for FXO the Web interface does not display this parameter. ▪ It is possible to configure whether the KeyBlindTransfer code is added as a prefix to the dialed destination number, by using the parameter KeyBlindTransferAddPrefix (refer to "Channel Parameters" on page 377).
Call Waiting (**Note:** The call waiting can be viewed in the 'Call Waiting' table - refer to "Call Waiting" on page 440.)	
Activate **[KeyCallWaiting]**	Keypad sequence that activates the Call Waiting option. After the sequence is pressed, a confirmation tone is heard.
Deactivate **[KeyCallWaitingDeact]**	Keypad sequence that deactivates the Call Waiting option. After the sequence is pressed, a confirmation tone is heard.
Reject Anonymous Call (**Note:** You can reject anonymous calls per FXS interface, using RejectAnonymousCallPerPort - refer to "Analog Telephony Parameters" on page 355.)	
Activate **[KeyRejectAnonymous Call]**	Keypad sequence that activates the reject anonymous call option, whereby the device rejects incoming anonymous calls. After the sequence is pressed, a confirmation tone is heard.
Deactivate **[KeyRejectAnonymous CallDeact]**	Keypad sequence that de-activates the reject anonymous call option. After the sequence is pressed, a confirmation tone is heard.

3.4.5.2.6 Stand-Alone Survivability

The 'SAS Configuration' page allows you to configure the device's Stand-Alone Survivability (SAS) feature. This feature is useful for providing a local backup via the PSTN in Small or Medium Enterprises (SME) that are serviced by IP Centrex services. In such environments, the enterprise's incoming and outgoing telephone calls (external and internal) are controlled by the Proxy, which communicates with the enterprise through the WAN interface. SAS ensures that incoming, outgoing, and internal calls service is maintained in case of a WAN or Proxy failure using a PSTN (or an alternate VoIP) backup connection and the device's built-in internal routing. To utilize the SAS feature, the VoIP CPEs such as IP phones or residential gateways, need to be defined so that their Proxy and Registrar destination addresses and UDP port equal the SAS feature's IP address and SAS local SIP UDP port.

Note: The SAS feature is applicable only to Mediant 1000. The 'SAS Configuration' page is Feature Key dependant and therefore is available only if included in the device's Feature Key (refer to 'Updating the Software Upgrade Key' on page 268).

The SAS feature operates in one of two modes:

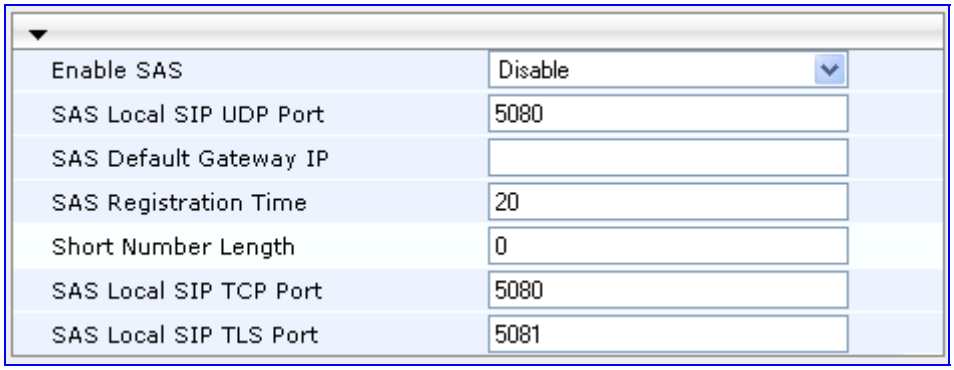
- **Normal Mode:** Initially, the SAS feature acts as a registrar so that every VoIP CPE within the SME's LAN registers to it, while the SAS feature passes all registration requests to the Proxy. In this mode, the SAS functions as a stateful proxy, passing all SIP requests received from the enterprise to the Proxy, and vice versa. In parallel, the SAS feature continuously maintains a 'Keep-Alive' handshake with the Proxy using SIP OPTIONS or re-INVITE messages.
- **Emergency Mode:** The SAS feature switches to Emergency mode if it determines (from the 'Keepalive' responses) that the connection with the Proxy is lost. This can occur due to Proxy server failure or WAN problems. In this mode, when the connection with the Proxy server is down, the SAS feature controls all internal calls within the enterprise. In the case of outgoing calls, the SAS feature forwards them to a local VoIP gateway (this can be the same gateway where the SAS feature resides or a separate analog or digital gateway). For PSTN fallback, the local VoIP gateway should be equipped with analog (FXO) lines or digital (E1/T1) trunk(s) for PSTN connectivity. In this way, the enterprise preserves its capability for internal and outgoing calls.

  When Emergency mode is active, the SAS feature continuously attempts to access the Proxy using the regular 'Keepalive' method. After the connection is re-established, the SAS feature switches to pre-Normal mode. In this mode, the SAS feature maintains all terminations of existing calls while any new SIP call signaling (issued by new INVITE sessions) is transacted to/from the Proxy server. This is accomplished using the SAS feature 's database of current active calls. After releasing all calls established during Emergency mode, the SAS feature can continue operating in Normal mode.

➢ To configure the Stand-Alone Survivability parameters, take these 4 steps:

1. Open the 'SAS Configuration' page (**Configuration** tab > **Protocol Configuration** menu > **SIP Advanced Parameters** submenu > **Stand-Alone Survivability** page item).

Figure 3-74: SAS Configuration Page



2. Configure the parameters according to the table below.
3. Click the **Submit** button to apply your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-40: Stand-Alone Survivability Parameters Description

Parameter	Description
Enable SAS **[EnableSAS]**	Enables the Stand-Alone Survivability (SAS) feature. ▪ **[0]** Disable Disabled (default) ▪ **[1]** Enable = SAS is enabled When enabled, the device receives the registration requests from different SIP entities in the local network and then forwards them to the defined proxy. If the connection to the proxy fails ('Emergency Mode'), the device serves as a proxy by allowing calls internal to the local network or outgoing to PSTN.
SAS Local SIP UDP Port **[SASLocalSIPUDPPort]**	Local UDP port for sending and receiving SIP messages for SAS. The SIP entities in the local network need to send the registration requests to this port. When forwarding the requests to the proxy ('Normal Mode'), this port serves as the source port. The valid range is 1 to 65,534. The default value is 5080.
SAS Default Gateway IP **[SASDefaultGatewayIP]**	The default gateway used in SAS 'Emergency Mode'. When an incoming SIP INVITE is received and the destination Address-Of-Record is not included in the SAS database, the request is immediately sent to this default gateway. The address can be configured as an IP address (dotted-decimal notation) or as a domain name (up to 49 characters). The default is a null string, which is interpreted as the local IP address of the gateway.
SAS Registration Time **[SASRegistrationTime]**	Determines the value of the SIP Expires header that is sent in a 200 OK response to an incoming REGISTER message when in SAS 'Emergency Mode'. The valid range is 0 (Analog) or 10 (Digital) to 2,000,000. The default value is 20.

Parameter	Description
Short Number Length **[SASShortNumberLength]**	Determines the length of the suffix used for SAS Short Numbering support. The dialed number is compared to the registered number. The registered (full) and dialed (full or short) numbers match only if one of the following conditions is met: ▪ The numbers match entirely (i.e., they are of the same length and all characters match). ▪ The dialed number is shorter than the registered one, its length equals to the length specified by the parameter and it equals the suffix of the registered number determined by the length in the parameter. When set to 0, only a full match between the dialed and registered numbers is valid. The valid range is 0 to 63 characters. The default value is no character. **Note:** This logic is only relevant in SAS Emergency mode.
SAS Local SIP TCP Port **[SASLocalSIPTCPPort]**	Local TCP port used to send/receive SIP messages for the SAS application. The SIP entities in the local network need to send the registration requests to this port. When forwarding the requests to the proxy ('Normal Mode'), this port serves as the source port. The valid range is 1 to 65,534. The default value is 5080.
SAS Local SIP TLS Port **[SASLocalSIPTLSPort]**	Local TLS port used to send/receive SIP messages for the SAS application. The SIP entities in the local network need to send the registration requests to this port. When forwarding the requests to the proxy ('Normal Mode'), this port serves as the source port. The valid range is 1 to 65,534. The default value is 5081.

3.4.5.3 Configuring the Number Manipulation Tables

The device provides four Number Manipulation tables for incoming (IP-to-Tel) and outgoing (Tel-to-IP) calls. These tables are used to modify the destination and source telephone numbers so that the calls can be routed correctly.

Telephone number manipulation can be implemented for the following usage examples:

- Strip or add dialing plan digits from or to the number. For example, a user may need to first dial 9 before dialing the phone number to indicate an external line. This number (9) can then be removed (by the Manipulation table) before the call is setup.
- Allow or disallow Caller ID information to be sent according to destination or source prefixes. For detailed information on Caller ID, refer to Caller Display Information on page 221.
- For digital modules only: Assign NPI/TON to IP-to-Tel calls. The device can use a single global setting for NPI/TON classification or it can use the setting in this table on a call-by-call basis.

The number manipulation is performed in the following tables:

- Destination Phone Number Manipulation Table for IP-to-Tel Calls (NumberMapIP2Tel *ini* file parameter)
- Destination Phone Number Manipulation Table for Tel-to-IP Calls (NumberMapTel2IP *ini* file parameter)
- Source Phone Number Manipulation Table for IP-to-Tel Calls (SourceNumberMapIP2Tel *ini* file parameter)
- Source Phone Number Manipulation Table for Tel-to-IP Calls (SourceNumberMapTel2IP *ini* file parameter)

Notes:

- Number manipulation can occur before or after a routing decision is made. For example, you can route a call to a specific Hunt (analog module) or Trunk (digital module) Group according to its original number, and then you can remove or add a prefix to that number before it is routed. To determine when number manipulation is performed, configure the 'IP to Tel Routing Mode' parameter (RouteModeIP2Tel) described in "IP to Trunk Group Routing" on page 204, and 'Tel to IP Routing Mode' parameter (RouteModeTel2IP) described in "Tel to IP Routing Table" on page 200.
- For configuring number manipulation using *ini* file table parameters NumberMapIP2Tel, NumberMapTel2IP, SourceNumberMapIP2Tel, and SourceNumberMapTel2IP, refer to "Number Manipulation and Routing Parameters" on page 364.

➢ To configure the Number Manipulation tables, take these 5 steps:

1. Open the required 'Number Manipulation' page (**Configuration** tab > **Protocol Configuration** menu > **Manipulation Tables** submenu > **Dest Number IP->Tel**, **Dest Number Tel->IP**, **Source Number IP->Tel**, or **Source Number Tel->IP** page item); the relevant Manipulation table page is displayed (e.g., 'Source Phone Number Manipulation Table for Tel→IP Calls' page).

Figure 3-75: Source Phone Number Manipulation Table for Tel-to-IP Calls

	Destination Prefix	Source Prefix	Stripped Digits Number	Prefix (Suffix) to Add	Number of Digits to Leave	Presentation
1	03	201	0	971		Allowed
2		1001	4	5(23)		Restricted
3		123451001#	0	(8)	4	Not Configured
4		[30-40]xx	(1)	2		Not Configured
5	[6,7,8]	2001	5	3		Not Configured
6						Not Configured

The figure above shows an example of the use of manipulation rules in the 'Source Phone Number Manipulation Table for Tel→IP Calls':

- When the destination number is 035000 and source number is 20155, the source number is changed to 97120155.
- When the source number is 1001876, it is changed to 587623.

- When the source number is 1234510012001, it is changed to 20018.
- When the source number is 3122, it is changed to 2312.

2. From the 'Table Index' drop-down list, select the range of entries that you want to edit (up to 20 entries can be configured for Source Number Manipulation and 50 entries for Destination Number Manipulation).
3. Configure the Number Manipulation table according to the table below.
4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Notes:

- The manipulation rules are executed in the following order:
  1. Number of stripped digits.
  2. Number of digits to leave.
  3. Prefix / suffix to add.
- The manipulation rules are applied to any incoming call whose:
  1. Destination number prefix matches the prefix defined in the 'Destination Prefix' field.
  2. Source number prefix matches the prefix defined in the 'Source Prefix' field.
  3. Source IP address matches the IP address defined in the 'Source IP' field (if applicable).

  The number manipulation can be performed using a combination of each of the above criteria, or using each criterion independently.
- For available notations that represent multiple numbers, refer to "Dialing Plan Notation" on page 194.

Table 3-41: Number Manipulation Parameters Description

Parameter	Description
Destination Prefix	Destination (called) telephone number prefix. An asterisk (*) represents any number.
Source Prefix	Source (caller) telephone number prefix. An asterisk (*) represents any number.
Source IP	Source IP address of the caller (obtained from the Contact header in the INVITE message). **Notes:** ▪ This parameter is only applicable to the Number Manipulation tables for IP-to-Tel calls. ▪ The source IP address can include the 'x' wildcard to represent single digits. For example: 10.8.8.xx represents all IP addresses between 10.8.8.10 to 10.8.8.99. ▪ The source IP address can include the asterisk (*) wildcard to represent any number between 0 and 255. For example, 10.8.8.* represents all IP addresses between 10.8.8.0 and 10.8.8.255.

Parameter	Description
Number of Stripped Digits	▪ Number of digits removed from the left of the telephone number prefix: Enter the number of digits that you want removed. For example, if you enter 3 and the phone number is 5551234, the new phone number is 1234. ▪ Number of digits removed from the right of the telephone number prefix: Enter the number of digits in parenthesis (). **Note:** A combination of the two options is allowed, e.g., 2(3).
Prefix (Suffix) to Add	▪ Prefix: Enter the number or string you want added to the front of the phone number. For example, if you enter 9 and the phone number is 1234, the new number is 91234. ▪ Suffix: Enter the number or string in brackets you want added to the end of the phone number. For example, if you enter (00) and the phone number is 1234, the new number is 123400. **Note:** A combination of the two options is allowed, e.g., 9(00).
Number of Digits to Leave	Enter the number of digits that you want to retain from the right of the phone number.
NPI	The Numbering Plan Indicator (NPI) assigned to this entry. ▪ **[0]** Unknown (default) ▪ **[9]** Private ▪ **[1]** E.164 Public ▪ **[-1]** Not Configured = value received from PSTN/IP is used **Notes:** ▪ This parameter is applicable only to Number Manipulation tables for IP-to-Tel calls. ▪ For a detailed list of the available NPI/TON values, refer to Numbering Plans and Type of Number on page 195
TON	The Type of Number (TON) assigned to this entry. ▪ If you selected 'Unknown' for the NPI, you can select Unknown [0]. ▪ If you selected 'Private' for the NPI, you can select Unknown [0], Level 2 Regional [1], Level 1 Regional [2], PISN Specific [3] or Level 0 Regional (Local) [4]. ▪ If you selected 'E.164 Public' for the NPI, you can select Unknown [0], International [1], National [2], Network Specific [3], Subscriber [4] or Abbreviated [6]. **Notes:** ▪ This parameter is applicable only to Number Manipulation tables for IP-to-Tel calls. ▪ The default is 'Unknown'.

Parameter	Description
Presentation	Determines whether Caller ID is permitted: ▪ Allowed = sends Caller ID information when a call is made using these destination / source prefixes. ▪ Restricted = restricts Caller ID information for these prefixes. ▪ Not Configured = privacy is determined according to the Caller ID table (refer to "Caller ID" on page 221). **Notes:** ▪ Only applicable to Number Manipulation tables for source number manipulation. ▪ If 'Presentation' is set to 'Restricted' and 'Asserted Identity Mode' is set to 'P-Asserted', the From header in the INVITE message includes the following: From: 'anonymous' <sip: anonymous@anonymous.invalid> and 'privacy: id' header.

3.4.5.3.1 Dialing Plan Notation

The dialing plan notation applies to all the Number Manipulation tables as well as to the 'Tel to IP Routing' table (refer to "Tel to IP Routing Table" on page 200) and to the 'IP to Trunk Group Routing' table (refer to "IP to Trunk Group Routing" on page 204). The dialing notation applies to digits entered for the destination and source prefixes to represent multiple numbers.

Table 3-42: Dialing Plan Notations

Notation	Description	Example
[n-m]	Represents a range of numbers. **Note:** Range of letters is not supported.	▪ [5551200-5551300]#: Represents all numbers from 5551200 to 5551300. ▪ 123[100-200]#: Represents all numbers from 123100 to 123200.
[n,m]	Represents multiple numbers. **Note:** This notation only supports single-digit numbers.	[2,3,4]xxx#: Represents four-digit numbers that start with 2, 3 or 4.
x	Represents any single digit.	54324: Represents any number that starts with 54324.
Pound sign (#) at the end of a number	Represents the end of a number.	54324xx#: Represents a 7-digit number that starts with 54324.
A single asterisk (*)	Represents any number.	*: Represents any number.

The device matches the rules starting at the top of the table (i.e., top rules take precedence over lower rules). For this reason, enter more specific rules above more generic rules. For example, if you enter 551 in entry 1 and 55 in entry 2, the device applies rule 1 to numbers that starts with 551 and applies rule 2 to numbers that start with 550, 552, 553, 554, 555, 556, 557, 558 and 559. However if you enter 55 in entry 1 and 551 in entry 2, the device applies rule 1 to all numbers that start with 55 including numbers that start with 551.

3.4.5.3.2 Numbering Plans and Type of Number

Numbers are classified by their Numbering Plan Indication (NPI) and their Type of Number (TON). The device supports all NPI/TON classifications used in the standard. The list of ISDN ETSI NPI/TON values is shown in the following table:

Table 3-43: NPI/TON Values for ISDN ETSI

NPI	TON	Description
Unknown [0]	Unknown [0]	A valid classification, but one that has no information about the numbering plan.
E.164 Public [1]	Unknown [0]	A public number in E.164 format, but no information on what kind of E.164 number.
	International [1]	A public number in complete international E.164 format, e.g., 16135551234.
	National [2]	A public number in complete national E.164 format, e.g., 6135551234.
	Subscriber [4]	A public number in complete E.164 format representing a local subscriber, e.g., 5551234.
Private [9]	Unknown [0]	A private number, but with no further information about the numbering plan.
	Level 2 Regional [1]	
	Level 1 Regional [2]	A private number with a location, e.g., 3932200.
	PISN Specific [3]	
	Level 0 Regional (local) [4]	A private local extension number, e.g., 2200.

For NI-2 and DMS-100 ISDN variants, the valid combinations of TON and NPI for calling and called numbers include (Plan/Type):

- 0/0 - Unknown/Unknown
- 1/1 - International number in ISDN/Telephony numbering plan
- 1/2 - National number in ISDN/Telephony numbering plan
- 1/4 - Subscriber (local) number in ISDN/Telephony numbering plan
- 9/4 - Subscriber (local) number in Private numbering plan

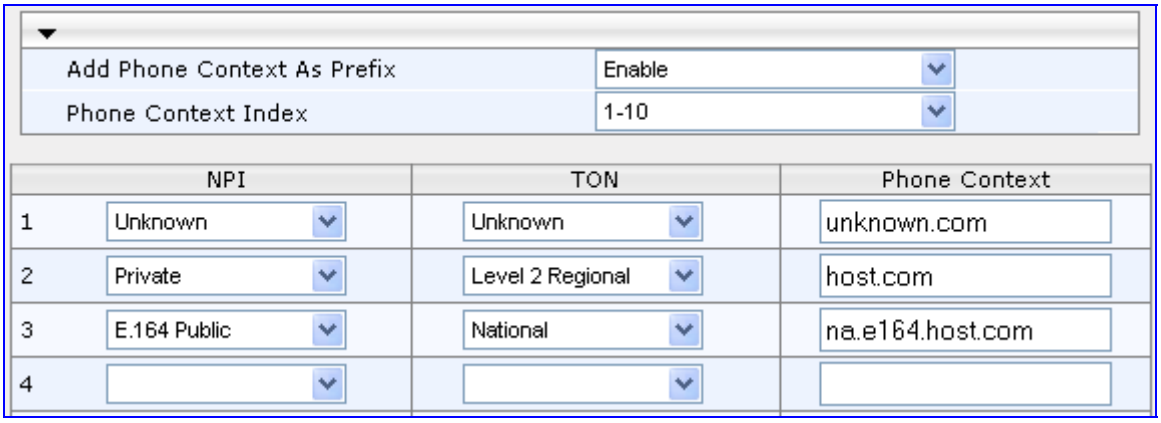
3.4.5.3.3 Mapping NPI/TON to Phone-Context

The 'Phone-Context Table' page is used to map NPI and TON to the Phone-Context SIP parameter. When a call is received from the ISDN/Tel, the NPI and TON are compared against the table and the Phone-Context value is used in the outgoing SIP INVITE message. The same mapping occurs when an INVITE with a Phone-Context attribute is received. The Phone-Context parameter appears in the standard SIP headers where a phone number is used (Request-URI, To, From, Diversion).

➢ To configure the Phone-Context tables, take these 4 steps:

1. Open the 'Phone Context Table' page (**Configuration** tab > **Protocol Configuration** menu > **Manipulation Tables** submenu > **Phone Context Table** page item).

Figure 3-76: Phone Context Table Page



2. Configure the Phone Context table according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Notes:

- Several rows with the same NPI-TON or Phone-Context are allowed. In such a scenario, a Tel-to-IP call uses the first match.
- Phone-Context '+' is a unique case as it doesn't appear in the Request-URI as a Phone-Context parameter. Instead, it's added as a prefix to the phone number. The '+' isn't removed from the phone number in the IP-to-Tel direction.
- You can also configure the Phone Context table using the *ini* file table parameter PhoneContext (refer to "Number Manipulation and Routing Parameters" on page 364).

Table 3-44: Phone-Context Parameters Description

Parameter	Description
Add Phone Context As Prefix **[AddPhoneContextAsPrefix]**	Determines whether the received Phone-Context parameter is added as a prefix to the outgoing ISDN SETUP message with Called and Calling numbers. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Enable = Enable.
NPI	Select the Number Plan assigned to this entry. ▪ **[0]** Unknown = Unknown (default) ▪ **[1]** E.164 Public = E.164 Public ▪ **[9]** Private = Private For a detailed list of the available NPI/TON values, refer to Numbering Plans and Type of Number on page 195.

Parameter	Description
TON	Select the Number Type assigned to this entry. ▪ If you selected Unknown as the NPI, you can select Unknown **[0]**. ▪ If you selected Private as the NPI, you can select Unknown **[0]**, Level 2 Regional **[1]**, Level 1 Regional **[2]**, PSTN Specific **[3]**, or Level 0 Regional (Local) **[4]**. ▪ If you selected E.164 Public as the NPI, you can select Unknown **[0]**, International **[1]**, National **[2]**, Network Specific **[3]**, Subscriber **[4]**, or Abbreviated **[6]**.
Phone Context	The Phone-Context SIP URI parameter.

3.4.5.4 Configuring the Routing Tables

The **Routing Tables** submenu allows you to configure the device's call routing. This submenu includes the following page items:

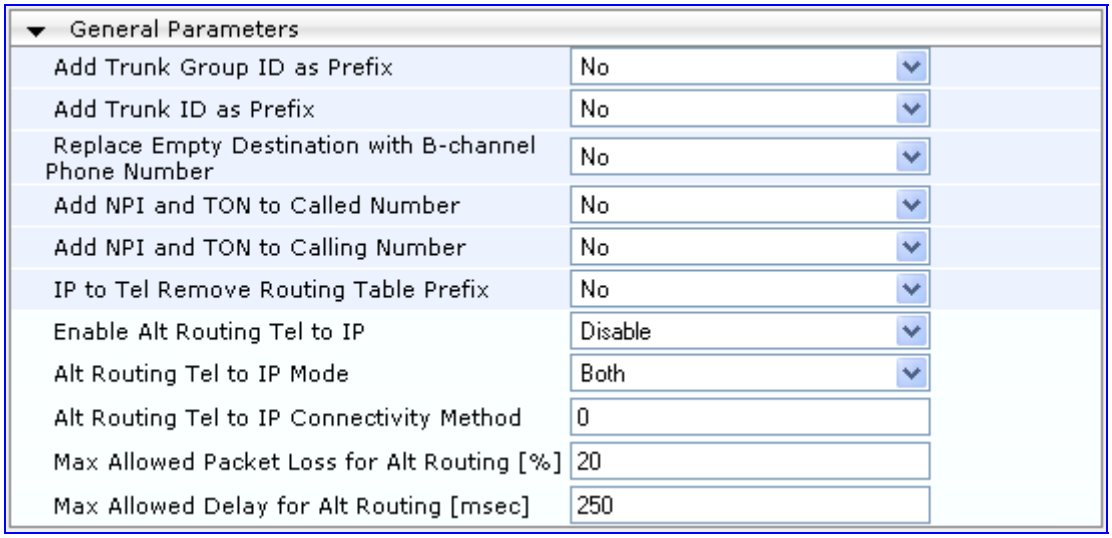
- Routing General Parameters (refer to "Routing General Parameters" on page 198)
- Tel to IP Routing (refer to "Tel to IP Routing Table" on page 200)
- IP to Trunk Group Routing (refer to "IP to Trunk Group Routing" on page 204)
- Internal DNS Table (refer to "Internal DNS Table" on page 207)
- Internal SRV Table (refer to "Internal SRV Table" on page 208)
- Reasons for Alternative Routing (refer to "Reasons for Alternative Routing" on page 209)
- Release Cause Mapping (refer to Release Cause Mapping on page 211)

3.4.5.4.1 Routing General Parameters

The 'Routing General Parameters' page allows you to configure the device's IP-to-Tel and Tel-to-IP routing parameters.

➢ **To configure the general routing parameters, take these 4 steps:**

1. Open the 'Routing General Parameters' page (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **Routing General Parameters** page item).



2. Configure the general parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-45: Routing General Parameters Description

Parameter	Description
Add Trunk Group ID as Prefix **[AddTrunkGroupAsPrefix]**	Determines whether the device's Trunk Group ID is added as a prefix to the destination phone number for Tel-to-IP calls. ▪ **[0]** No = Don't add Trunk Group ID as prefix (default). ▪ **[1]** Yes = Add Trunk Group ID as prefix to called number. **Notes:** ▪ This option can be used to define various routing rules. ▪ To use this feature, you must configure the Trunk Group IDs (refer to "Configuring the Trunk Group Table" on page 226).
Add Trunk ID as Prefix **[AddPortAsPrefix]**	Determines whether the Trunk ID is added as a prefix to the called number for Tel-to-IP calls. ▪ **[0]** No = Don't add Trunk ID as prefix (default). ▪ **[1]** Yes = Enable add Trunk ID as prefix. If enabled, the Trunk ID (single digit in the range 1 to 8) is added as a prefix to the called (destination) phone number. This option can be used to define various routing rules.

Parameter	Description
Replace Empty Destination with B-channel Phone Number **[ReplaceEmptyDstWithPortNumber]**	Determines whether the internal channel number is used as the destination number if the called number is missing. ▪ **[0]** No (default) ▪ **[1]** Yes **Note:** Applicable only for Tel-to-IP calls and if the called number is missing.
Add NPI and TON to Calling Number **[AddNPIandTON2CallingNumber]**	Determines whether Numbering Plan Indicator (NPI) and Type of Numbering (TON) are added to the Calling Number for Tel-to-IP calls. ▪ **[0]** No = Do not change the Calling Number (default). ▪ **[1]** Yes = Add NPI and TON to the Calling Number ISDN Tel-to-IP call. For example: After receiving a Calling Number of 555, NPI of 1, and TON of 3, the modified number becomes 13555. This number can later be used for manipulation and routing.
Add NPI and TON to Called Number **[AddNPIandTON2CalledNumber]**	Determines whether NPI and TON are added to the Called Number for Tel-to-IP calls. ▪ **[0]** No = Do not change the Called Number (default). ▪ **[1]** Yes = Add NPI and TON to the Called Number of ISDN Tel-to-IP call. For example: After receiving a Called Number of 555, NPI of 1 and TON of 3, the modified number becomes 13555. This number can later be used for manipulation and routing.
IP to Tel Remove Routing Table Prefix **[RemovePrefix]**	Determines whether the device removes the prefix from the destination number for IP-to-Tel calls. ▪ **[0]** No = Don't remove prefix (default) ▪ **[1]** Yes = Remove the prefix (defined in the 'IP to Trunk Group Routing' table - refer to "IP to Trunk Group Routing" on page 204) from a telephone number for an IP-to-Tel call, before forwarding it to Tel. For example: To route an incoming IP-to-Tel call with destination number 21100, the 'IP to Trunk Group Routing' table is scanned for a matching prefix. If such a prefix is found (e.g., 21), then before the call is routed to the corresponding Trunk Group, the prefix (21) is removed from the original number, and therefore, only 100 remains. **Notes:** ▪ Applicable only if number manipulation is performed after call routing for IP-to-Tel calls (i.e., RouteModeIP2Tel parameter is set to 0). ▪ Similar operation (of removing the prefix) is also achieved by using the usual number manipulation rules.
Enable Alt Routing Tel to IP **[AltRoutingTel2IPEnable]**	Enables the Alternative Routing feature for Tel-to-IP calls. ▪ **[0]** Disable = Disables the Alternative Routing feature (default). ▪ **[1]** Enable = Enables the Alternative Routing feature. ▪ **[2]** Status Only = The Alternative Routing feature is disabled, but read-only information on the Quality of Service of the destination IP addresses is provided. For information on the Alternative Routing feature, refer to "Configuring Alternative Routing (Based on Connectivity and QoS)" on page 415.

Parameter	Description
Alt Routing Tel to IP Mode **[AltRoutingTel2IPMode]**	Determines the event(s) reason for triggering Alternative Routing. ▪ **[0]** None = Alternative routing is not used. ▪ **[1]** Connectivity = Alternative routing is performed if ping to initial destination fails. ▪ **[2]** QoS = Alternative routing is performed if poor QoS is detected. ▪ **[3]** Both = Alternative routing is performed if either ping to initial destination fails, poor Quality of Service is detected, or DNS host name is not resolved (default). **Notes:** ▪ QoS is quantified according to delay and packet loss calculated according to previous calls. QoS statistics are reset if no new data is received within two minutes. For information on the Alternative Routing feature, refer to "Configuring Alternative Routing (Based on Connectivity and QoS)" on page 415. ▪ To receive quality information (displayed in the 'Quality Status' and 'Quality Info.' fields in "IP Connectivity" on page 289) per destination, this parameter must be set to 2 or 3.
Alt Routing Tel to IP Connectivity Method **[AltRoutingTel2IPConnMethod]**	Determines the method used by the device for periodically querying the connectivity status of a destination IP address. ▪ **[0]** ICMP Ping (default) = Internet Control Message Protocol (ICMP) ping messages. ▪ **[1]** SIP OPTIONS = The remote destination is considered offline if the latest OPTIONS transaction timed out. Any response to an OPTIONS request, even if indicating an error, brings the connectivity status to online.
Max Allowed Packet Loss for Alt Routing [%] **[IPConnQoSMaxAllowedPL]**	Packet loss percentage at which the IP connection is considered a failure and Alternative Routing mechanism is activated. The range is 1 to 20%. The default value is 20%.
Max Allowed Delay for Alt Routing [msec] **[IPConnQoSMaxAllowedDelay]**	Transmission delay (in msec) at which the IP connection is considered a failure and Alternative Routing mechanism is activated. The range is 100 to 1000. The default value is 250.

3.4.5.4.2 Tel to IP Routing Table

The 'Tel to IP Routing' page provides a table for routing Tel calls to IP addresses. This routing table associates a called / calling telephone number's prefixes with a destination IP address or with a Fully Qualified Domain Name (FQDN). When a call is routed by the device (i.e., a Proxy server isn't used), the called and calling numbers are compared to the list of prefixes in this table. Calls that match these prefixes are sent to the corresponding IP address. If the number dialed does not match these prefixes, the call is not made. You can add up to 50 Tel-to-IP routing rules in the table.

When using a Proxy server, you do not need to configure this table, unless you require one of the following:

- Fallback routing when communication with Proxy servers is lost.
- Implement the 'Filter Calls to IP' and 'IP Security' features.
- Obtain different SIP URI host names (per called number).
- Assign IP profiles.

Note that for this table to take precedence over a Proxy for routing calls, set the parameter PreferRouteTable to 1. The device checks the 'Destination IP Address' field in this table for a match with the outgoing call. A Proxy is used only if a match is not found.

Possible uses for Tel-to-IP routing include the following:

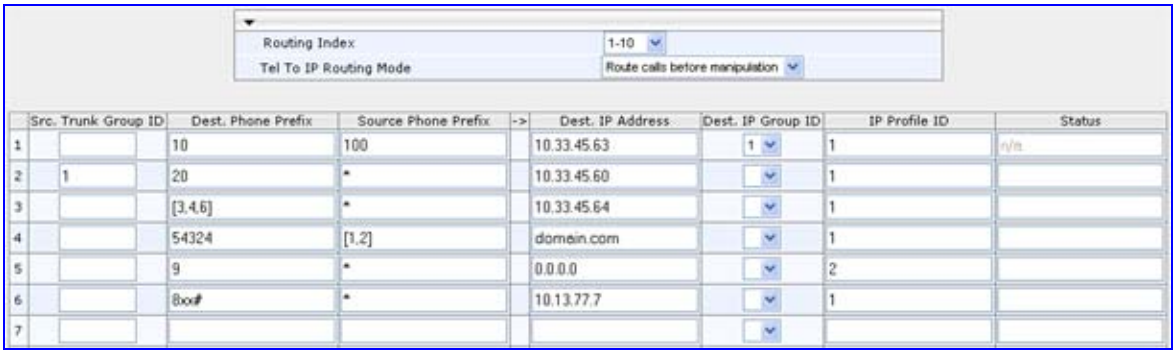
- Fallback to internal routing table if there is no communication with the Proxy servers.
- Call Restriction (when Proxy isn't used): rejects all outgoing Tel-to-IP calls that are associated with the destination IP address 0.0.0.0.
- IP Security: When the IP Security feature is enabled (SecureCallFromIP = 1), the device accepts only those IP-to-Tel calls with a source IP address defined in the 'Tel to IP Routing' table.
- Filter Calls to IP: When a Proxy is used, the device checks the 'Tel to IP Routing' table before a telephone number is routed to the Proxy. If the number is not allowed (number isn't listed or a Call Restriction routing rule is applied), the call is released.
- Always Use Routing Table: When this feature is enabled (AlwaysUseRouteTable = 1), even if a Proxy server is used, the SIP URI host name in the sent INVITE message is obtained from this table. Using this feature, you can assign a different SIP URI host name for different called and/or calling numbers.
- Assign Profiles to destination addresses (also when a Proxy is used).
- Alternative Routing (when a Proxy isn't used): an alternative IP destination for telephone number prefixes is available. To associate an alternative IP address to called telephone number prefix, assign it with an additional entry (with a different IP address), or use an FQDN that resolves into two IP addresses. The call is sent to the alternative destination when one of the following occurs:
  - No ping to the initial destination is available, poor QoS (delay or packet loss, calculated according to previous calls) is detected, or a DNS host name is not resolved. For detailed information on Alternative Routing, refer to "Configuring Alternative Routing (Based on Connectivity and QoS" on page 415.
  - A release reason defined in the 'Reasons for Alternative Tel to IP Routing' table is received (refer to "Reasons for Alternative Routing" on page 209).

Alternative routing (using this table) is commonly implemented when there is no response to an INVITE message (after INVITE retransmissions). The device then issues an internal 408 'No Response' implicit release reason. If this reason is included in the 'Reasons for Alternative Routing' table, the device immediately initiates a call to the redundant destination using the next matched entry in the 'Tel to IP Routing' table. Note that if a domain name in this table is resolved into two IP addresses, the timeout for INVITE retransmissions can be reduced by using the parameter 'Number of RTX Before Hotswap'.

➢ To configure the Tel to IP Routing table, take these 6 steps:

1. Open the 'Tel to IP Routing' page (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **Tel to IP Routing** page item).

Figure 3-77: Tel to IP Routing Page



2. From the 'Routing Index' drop-down list, select the range of entries that you want to add.
3. From the 'Tel to IP Routing Mode' drop-down list, select the required Tel to IP routing mode regarding number manipulation (refer to the table below).
4. Configure the Tel to IP Routing table according to the table below.
5. Click the **Submit** button to save your changes.
6. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Notes:

- If the alternative routing destination is the device itself, the call can be configured to be routed back to the PSTN. This feature is referred to as 'PSTN Fallback', meaning that if poor voice quality occurs over the IP network, the call is routed through the legacy telephony system (PSTN).
- Tel-to-IP routing can be performed before or after applying the number manipulation rules. To control when number manipulation is performed, use the 'Tel to IP Routing Mode' (or RouteModeTel2IP *ini* file) parameter, described in the table below.
- You can also configure the 'Tel to IP Routing' table using the *ini* file table parameter Prefix (refer to "Number Manipulation and Routing Parameters" on page 364).

Table 3-46: Tel to IP Routing Table Parameters Description

Parameter	Description
Tel to IP Routing Mode **[RouteModeTel2IP]**	Determines whether to route Tel calls to IP before or after manipulation of destination number. ▪ **[0]** Route calls before manipulation = Tel-to-IP calls are routed before the number manipulation rules are applied (default). ▪ **[1]** Route calls after manipulation = Tel-to-IP calls are routed after the number manipulation rules are applied. **Notes:** Not applicable if outbound Proxy routing is used.

Parameter	Description
Src. Trunk Group ID	The source Trunk Group for Tel-to-IP calls. The range is 1-99. **Notes:** ▪ If this parameter is not required in the routing rule, leave the field empty. ▪ To denote any Trunk Group, you can enter the asterisk (*) symbol.
Dest. Phone Prefix	Represents a called telephone number prefix. The prefix can be 1 to 19 digits long. An asterisk (*) represents all numbers.
Source Phone Prefix	Represents a calling telephone number prefix. The prefix can be 1 to 19 digits long. An asterisk (*) represents all numbers.
Any telephone number whose destination number matches the prefix defined in the 'Dest. Phone Prefix' field and whose source number matches the prefix defined in the 'Source Phone Prefix' field is sent to the IP address defined in the 'Dest. IP Address' field. **Notes:** ▪ Tel-to-IP routing can be performed according to a combination of source and destination phone prefixes, or using each independently. ▪ An additional entry of the same prefixes can be assigned to enable alternative routing. ▪ For available notations that represent multiple numbers, refer to "Dialing Plan Notation" on page 194.	
Dest. IP Address	The IP address (and optionally port number) assigned to the prefix. For example, <IP address>:<Port>. Domain names such as domain.com can be used instead of IP addresses. To discard outgoing IP calls, enter 0.0.0.0. The IP address 127.0.0.1 can be used when the IP address of the device itself is unknown (for example, when DHCP is used). **Note:** When using domain names, you must enter a DNS server IP address or alternatively, define these names in the 'Internal DNS Table' (refer to "Internal DNS Table" on page 207).
Dest IP Group ID	The IP Group (1-9) to where you want to route the Tel-to-IP call. The INVITE messages are sent to the IP address(es) of the Proxy Set that is associated with the selected IP Group. Typically, if you select an IP Group, it is unnecessary to configure a destination IP address (in the 'Dest IP Address' field). However, if both parameters are configured, the INVITE message is sent only to the IP Group. If the parameter AlwaysUseRouteTable is set to 1 (in the IP Group table, refer to "Configuring the IP Groups" on page 231), the request URI host name in the INVITE message is set to the value of the parameter 'Dest IP Address' (if not empty); otherwise, it is set to the value of the parameter 'SIP Group Name' (defined in the IP Group table). **Note:** To configure Proxy Sets, refer to "Proxy Sets Table" on page 153.
IP Profile ID	IP profile number assigned to the destination IP address that is defined in the 'Dest. IP Address' field.

Parameter	Description
Status	A read-only field representing the Quality of Service of the destination IP address: ▪ n/a = Alternative Routing feature is disabled. ▪ OK = IP route is available. ▪ Ping Error = No ping to IP destination; route is not available. ▪ QoS Low = Bad QoS of IP destination; route is not available. ▪ DNS Error = No DNS resolution (only when domain name is used instead of an IP address).
Charge Code	An optional Charge Code (1 to 25) can be applied to each routing rule to associate it with an entry in the Charge Code table (refer to "Charge Codes Table" on page 183). **Note:** This column is applicable only to Mediant 1000.

3.4.5.4.3 IP to Trunk Group Routing

The 'IP to Trunk Group Routing Table' page provides a table for routing incoming IP calls to groups of channels (BRI,E1/T1 B-channels, and Mediant 1000 FXS/FXO endpoints) called Trunk Groups. You can add up to 24 entries in the table.

The IP-to-Tel calls are routed to Trunk Groups according to any of the following (or a combination thereof):

- Destination phone prefix
- Source phone prefix
- Source IP address

Once the call is routed to the specific Trunk Group, the call is sent to the device's channels pertaining to the Trunk Group. The specific channel within the Trunk Group to which the call is sent is determined according to the Trunk Group's channel selection mode. This mode is defined per Trunk Group in the 'Trunk Group Settings' page (refer to "Configuring the Trunk Group Settings" on page 228) or for all Trunk Groups in the 'SIP General Parameters' page (refer to "SIP General Parameters" on page 134) using the global parameter ChannelSelectMode. If a mode is not specified for a Trunk Group, then this global parameter applies. Trunk Groups can be used for FXO and FXS channels, but they are typically used for FXO interfaces (only Mediant 1000).

Trunk Group ID's are assigned to the device's channels in the 'Trunk Group Table' page (refer to "Configuring the Trunk Group Table" on page 226).

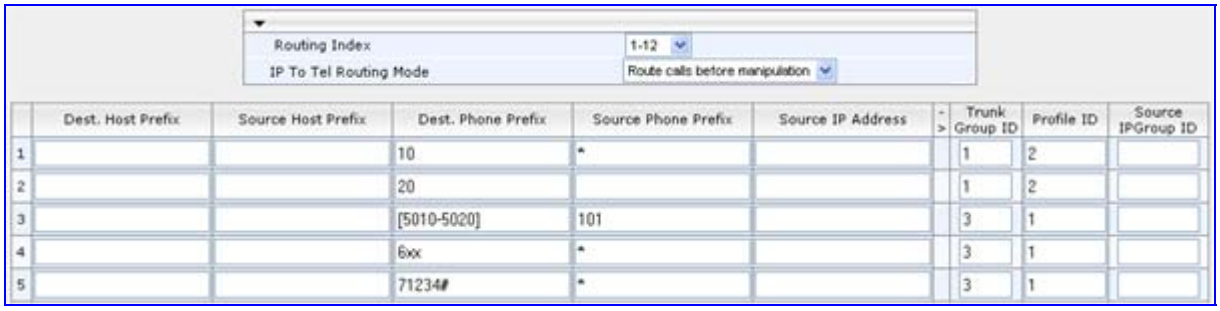
Notes:

- When a release reason, defined in the 'Reasons for Alternative IP to Tel Routing' table (refer to "Reasons for Alternative Routing" on page 209) is received for a specific IP-to-Tel call, an alternative Trunk Group for that call is available. To associate an alternative Trunk Group to an incoming IP call, assign it with an additional entry in the 'IP to Trunk Group Routing' table (repeat the same routing rules with a different Trunk Group ID).
- You can also configure the 'IP to Trunk Group Routing' table using the *ini* file table parameter PSTNPrefix (refer to "Number Manipulation and Routing Parameters" on page 364).

➢ To configure the IP to Trunk Group Routing table, take these 6 steps:

1. Open the 'IP to Trunk Group Routing' page (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **IP to** Trunk **Group Routing** page item).

Figure 3-78: IP to Trunk Group Routing Table Page



2. From the 'Routing Index' drop-down list, select the range of entries that you want to add.
3. From the 'IP to Tel Routing Mode' field, select the IP to Tel routing mode regarding number manipulation (refer to the table below).
4. Configure the table according to the table below.
5. Click the **Submit** button to save your changes.
6. To save the changes so they are available after a power failure, refer to "Saving Configuration" on page 265.

Table 3-47: IP to Trunk Group Routing Table Description

Parameter	Description
IP to Tel Routing Mode **[RouteModeIP2Tel]**	Determines whether to route IP calls to the Trunk group before or after manipulation of destination number. ▪ **[0]** Route calls before manipulation = IP-to-Tel calls are routed before the number manipulation rules are applied (default). ▪ **[1]** Route calls after manipulation = IP-to-Tel calls are routed after the number manipulation rules are applied. **Note:** To configure number manipulation, refer to "Configuring the Number Manipulation Tables" on page 190.
Dest. Host Prefix	The request URI host name prefix of the incoming INVITE message. If configured, the device uses it for IP-to-Tel routing. If not used, it must be left empty. **Note:** The asterisk (*) symbol cannot be used to depict any destination host prefix.

Parameter	Description
Source Host Prefix	The From URI host name prefix of the incoming INVITE message. If configured, the device uses it for IP-to-Tel routing. If not used, it must be left empty. **Notes:** ▪ The asterisk (*) symbol cannot be used to depict any source host prefix. ▪ If the P-asserted-ID header is present in the incoming INVITE message, then the parameter 'Source Host Prefix' is compared to the P-Asserted-ID URI hostname and not to the From header.
Dest. Phone Prefix	Represents a called telephone number prefix. The prefix can be 1 to 49 digits long. **Note:** An asterisk (*) represents all numbers.
Source Phone Prefix	Represents a calling telephone number prefix. The prefix can be 1 to 49 digits long. **Note:** An asterisk (*) represents all numbers.
Source IP Address	The source IP address of an IP-to-Tel call (obtained from the Contact header in the INVITE message) that can be used for routing decisions. **Notes:** ▪ You can configure from where the source IP address is taken, using the ini file parameter SourceIPAddressInput (refer to "Number Manipulation and Routing Parameters" on page 364). ▪ The source IP address can include the letter ex (x) wildcard to represent single digits. For example: 10.8.8.xx represents all the addresses between 10.8.8.10 to 10.8.8.99. ▪ The source IP address can include the asterisk (*) wildcard to represent any number between 0 and 255. For example, 10.8.8.* represents all addresses between 10.8.8.0 and 10.8.8.255.
Trunk Group ID	The Trunk Group to which calls are assigned that match the phone prefixes defined in the table. Any incoming SIP call whose destination prefix number, source prefix number, and source IP address matches those defined in the parameters described above is assigned to the corresponding Trunk Group ID. **Notes:** ▪ IP-to-Trunk Group routing can be performed according to any combination of source / destination phone prefixes and source IP address, or using each independently. ▪ For available notations that represent multiple numbers (used in the prefix columns), refer to "Dialing Plan Notation" on page 194.
Profile ID	IP profile assigned to the routing rule.
Source IP Group ID	Selects the source IP Group (1-9) associated with the incoming IP-to-Tel call. This is the IP Group from where the INVITE message was sent. This IP Group can later be used as the 'ServingIPGroup' in the Account table (refer to "Configuring the Account Table" on page 232) for obtaining authentication user name/password for this call.

3.4.5.4.4 Internal DNS Table

The 'Internal DNS Table' page, similar to a DNS resolution is used to translate up to 10 host (domain) names into IP addresses (e.g., when using the 'Tel to IP Routing' table). Two different IP addresses can be assigned to the same host name, typically used for alternative routing (using the 'Tel to IP Routing' table).

Notes:

- The device initially attempts to resolve a domain name using this Internal DNS table. If the domain name isn't listed in the table, the device performs a DNS resolution using an external DNS server.
- You can also configure the DNS table using the *ini* file table parameter DNS2IP (refer to "Networking Parameters" on page 296).

➢ To configure the internal DNS table, take these 6 steps:

1. Open the 'Internal DNS Table' page (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **Internal DNS Table** page item).

Figure 3-79: Internal DNS Table Page

	Domain Name	First IP Address	Second IP Address
1	DomainName.com	10.8.21.4	10.13.2.95
2			
3			
4			
5			
6			
7			
8			
9			
10			

2. In the 'Domain Name' field, enter the host name to be translated. You can enter a string of up to 31 characters long.
3. In the 'First IP Address' field, enter the first IP address (in dotted-decimal format notation) to which the host name is translated.
4. In the 'Second IP Address' field, enter the second IP address to which the host name is translated.
5. Click the **Submit** button to save your changes.
6. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

3.4.5.4.5 Internal SRV Table

The 'Internal SRV Table' page provides a table for resolving host names to DNS A-Records. Three different A-Records can be assigned to each host name. Each A-Record contains the host name, priority, weight, and port.

Notes:

- If the Internal SRV table is configured, the device initially attempts to resolve a domain name using this table. If the domain name isn't found, the device performs an Service Record (SRV) resolution using an external DNS server.
- You can also configure the Internal SRV table using the *ini* file table parameter SRV2IP (refer to "Networking Parameters" on page 296).

➢ To configure the Internal SRV table, take these 9 steps:

1. Open the 'Internal SRV Table' page (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **Internal SRV Table** page item).

Figure 3-80: Internal SRV Table Screen

2. In the 'Domain Name' field, enter the host name to be translated. You can enter a string of up to 31 characters long.
3. From the 'Transport Type' drop-down list, select a transport type.
4. In the 'DNS Name 1' field, enter the first DNS A-Record to which the host name is translated.
5. In the 'Priority', 'Weight' and 'Port' fields, enter the relevant values
6. Repeat steps 4 through 5, for the second and third DNS names, if required.
7. Repeat steps 2 through 6, for each entry.
8. Click the **Submit** button to save your changes.
9. To save the changes so they are available after a hardware reset or power fail, refer to "Saving Configuration" on page 265.

3.4.5.4.6 Reasons for Alternative Routing

The 'Reasons for Alternative Routing' page includes two groups - IP to Tel Reasons and Tel to IP Reasons. Each group allows you to define up to four different release reasons. If a call is released as a result of one of these reasons, the device tries to find an alternative route for that call. The release reason for IP-to-Tel calls is provided in Q.931 notation. The release reason for Tel-to-IP calls is provided in SIP 4xx, 5xx, and 6xx response codes. For Tel-to-IP calls, an alternative IP address is provided; for IP-to-Tel calls an alternative Hunt (Mediant 1000 analog modules) or Trunk (digital modules) Group is provided. Refer to "Tel to IP Routing Table" on page 200 for information on defining an alternative IP address; refer to "IP to Trunk Group Routing" on page 204 for information on defining an alternative Trunk Group.

You can use the 'Reasons for Alternative Routing' page for the following example scenarios:

- **Tel-to-IP calls:** when there is no response to an INVITE message (after INVITE retransmissions), the device issues an internal 408 'No Response' implicit release reason.
- **IP-to-Tel calls:** when the destination is busy and release reason #17 is issued or for other call releases that issue the default release reason (#3). Refer to DefaultReleaseCause in "Advanced Parameters" on page 166.

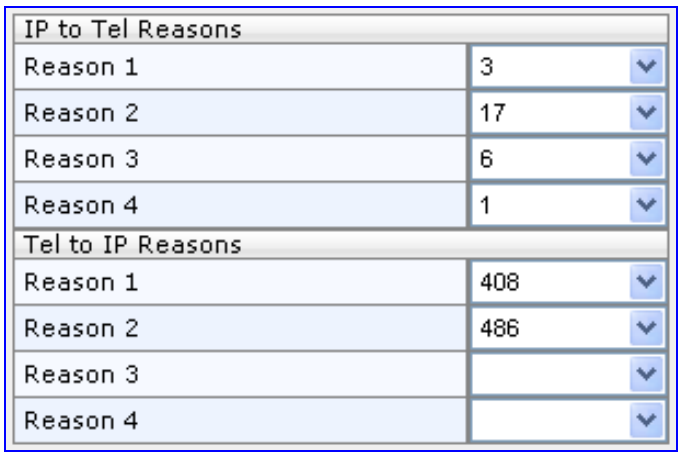
Notes:

- The reasons for alternative routing for Tel-to-IP calls only apply when a Proxy isn't used.
- For Tel-to-IP calls, the device sends the call to an alternative route only after the call has failed and the device has subsequently attempted twice to establish the call unsuccessfully.
- You can also configure alternative routing using the *ini* file table parameters AltRouteCauseTel2IP and AltRouteCauseIP2Tel (refer to "Number Manipulation and Routing Parameters" on page 364).

➢ To configure the reasons for alternative routing, take these 5 steps:

1. Open the 'Reasons for Alternative Routing' page (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **Reasons for Alternative Routing** page item).

Figure 3-81: Reasons for Alternative Routing Page



2. In the 'IP to Tel Reasons' group, select up to four different call failure reasons that invoke an alternative IP-to-Tel routing.
3. In the 'Tel to IP Reasons' group, select up to four different call failure reasons that invoke an alternative Tel-to-IP routing.
4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

3.4.5.4.7 Release Cause Mapping

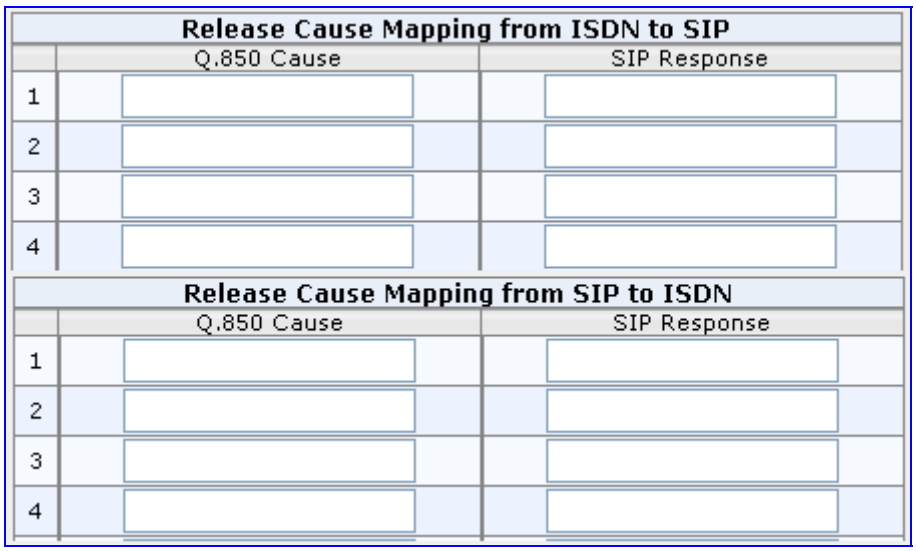
The 'Release Cause Mapping' page consists of two groups that allow the device to map up to 12 different SIP Responses to Q.850 Release Causes and vice versa, thereby overriding the hard-coded mapping mechanism (described in "Release Reason Mapping" on page 459).

Note: You can also configure SIP Responses-Q.850 Release Causes mapping using the *ini* file table parameters CauseMapISDN2SIP and CauseMapSIP2ISDN (refer to "ISDN and CAS Interworking-Related Parameters" on page 348).

➢ To configure Release Cause Mapping, take these 5 steps:

1. Open the 'Release Cause Mapping' page (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **Release Cause Mapping** page item).

Figure 3-82: Release Cause Mapping Page



2. In the 'Release Cause Mapping from ISDN to SIP' group, map (up to 12) different Q.850 Release Causes to SIP Responses.
3. In the 'Release Cause Mapping from SIP to ISDN' group, map (up to 12) different SIP Responses to Q.850 Release Causes.
4. Click the **Submit** button to save your changes.
5. To save the changes so they are available after a power failure, refer to "Saving Configuration" on page 265.

3.4.5.5 Configuring the Profile Definitions

The **Profile Definitions** submenu includes the following page items:

- Coder Group Settings (refer to "Coder Group Settings" on page 212)
- Tel Profile Settings (refer to "Tel Profile Settings" on page 214)
- IP Profile Settings (refer to "IP Profile Settings" on page 216)

Profile Definitions submenu allows you to implement the device's Profiles feature, which provides the device with high-level adaptation when connected to a variety of equipment (at both Tel and IP sides) and protocols, each of which requires different system behavior. You can assign different Profiles (behavior) per call, using the 'Tel to IP Routing' page (refer to "Tel to IP Routing Table" on page 200) and 'IP to Trunk Group Routing' page (refer to "IP to Trunk Group Routing" on page 204), or associate different Profiles per the device's channels.

Each Profile contains a set of parameters such as coders, T.38 Relay, Voice and DTMF Gain, Silence Suppression, Echo Canceler, RTP DiffServ, Current Disconnect and more. The Profiles feature allows you to customize these parameters or turn them on or off, per source or destination routing and/or per the device's trunks (channels). For example, specific E1/T1 spans can be assigned a Profile that always uses G.711.

Each call can be associated with one or two Profiles - Tel Profile and/or IP Profile. If both IP and Tel profiles apply to the same call, the coders and other common parameters of the preferred Profile (determined by the Preference option) are applied to that call. If the Preference of the Tel and IP Profiles is identical, the Tel Profile parameters take precedence.

Notes:

- The default values of the parameters in the 'Tel Profile Settings' and 'IP Profile Settings' pages are identical to their default values in their respective primary configuration page.
- If you modify a parameter in its primary configuration page (or *ini* file) that also appears in the profile pages, the parameter's new value is automatically updated in the profile pages. However, once you modify any parameter in the profile pages, modifications to parameters in the primary configuration pages (or *ini* file) no longer impact that profile pages.

3.4.5.5.1 Coder Group Settings

The 'Coder Group Settings' page provides a table for defining up to four different coder groups. These coder groups are used in the 'Tel Profile Settings' and 'IP Profile Settings' pages to assign different coders to Profiles.

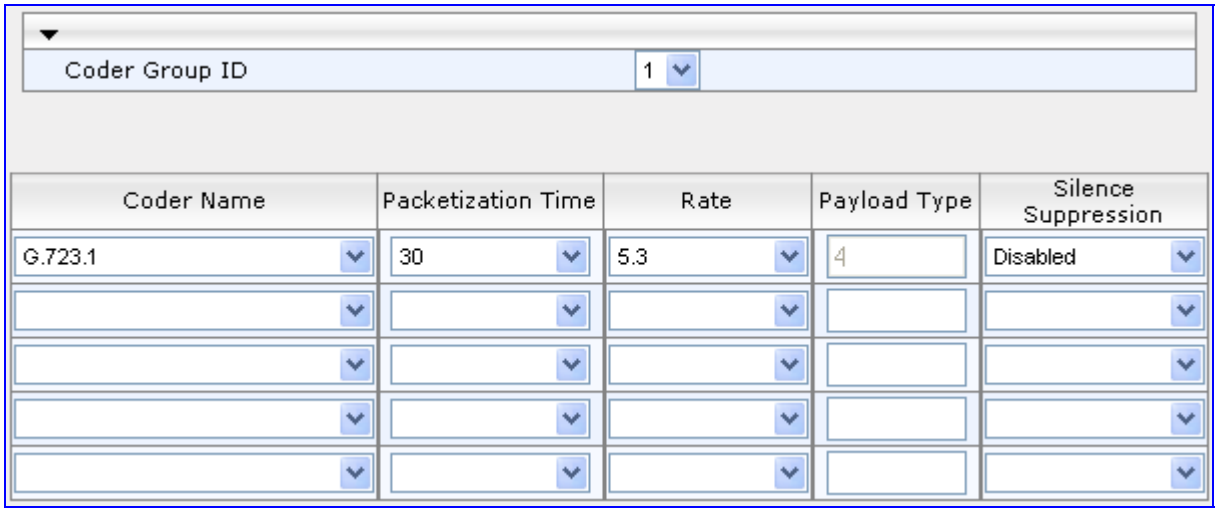
For each coder group you can define up to five coders, where the first coder (and its attributes) in the table takes precedence over the second coder, and so on. The first coder is the highest priority coder and is used by the device whenever possible. If the far end device cannot use the coder assigned as the first coder, the device attempts to use the next coder and so on. For a list of coders supported by the device, refer to "Coders" on page 157.

Note: You can also configure the coder groups using the *ini* file table parameter CoderName (refer to "SIP Configuration Parameters" on page 323).

➢ To configure coder groups, take these 11 steps:

1. Open the 'Coder Group Settings' page (**Configuration** tab > **Protocol Configuration** menu > **Profile Definitions** submenu > **Coder Group Settings** page item).

Figure 3-83: Coder Group Settings Page



2. From the 'Coder Group ID' drop-down list, select a coder group ID.
3. From the 'Coder Name' drop-down list, select the first coder for the coder group.
4. From the 'Packetization Time' drop-down list, select the packetization time (in msec) for the coder. The packetization time determines how many coder payloads are combined into a single RTP packet.
5. From the 'Rate' drop-down list, select the bit rate (in kbps) for the coder you selected.
6. In the 'Payload Type' field, if the payload type for the coder you selected is dynamic, enter a value from 0 to 120 (payload types of 'well-known' coders cannot be modified). The payload type identifies the format of the RTP payload.
7. From the 'Silence Suppression' drop-down list, enable or disable the silence suppression option for the coder you selected.
8. Repeat steps 3 through 7 for the second to fifth coders (optional).
9. Repeat steps 2 through 8 for the second to fourth coder groups (optional).
10. Click the **Submit** button to save your changes.
11. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Notes:

- Each coder can appear only once per Coder Group.
- The device always uses the packetization time requested by the remote side for sending RTP packets. If not specified, the packetization time (ptime) is assigned the default value.
- Only the packetization time of the first coder in the defined coder list is declared in INVITE / 200 OK SDP, even if multiple coders are defined.
- For G.729 it is also possible to select silence suppression without adaptations.
- If the coder G.729 is selected and silence suppression is enabled (for this coder), the device includes the string 'annexb=no' in the SDP of the relevant SIP messages. If silence suppression is set to 'Enable w/o Adaptations', 'annexb=yes' is included. An exception to this logic is when the remote device is a Cisco gateway (IsCiscoSCEMode).

3.4.5.5.2 Tel Profile Settings

The 'Tel Profile Settings' page allows you to define up to nine different Tel Profiles. You can then assign these Tel Profiles to the device's channels (in the 'Trunk Group Table' page), thereby applying different behaviors to different channels.

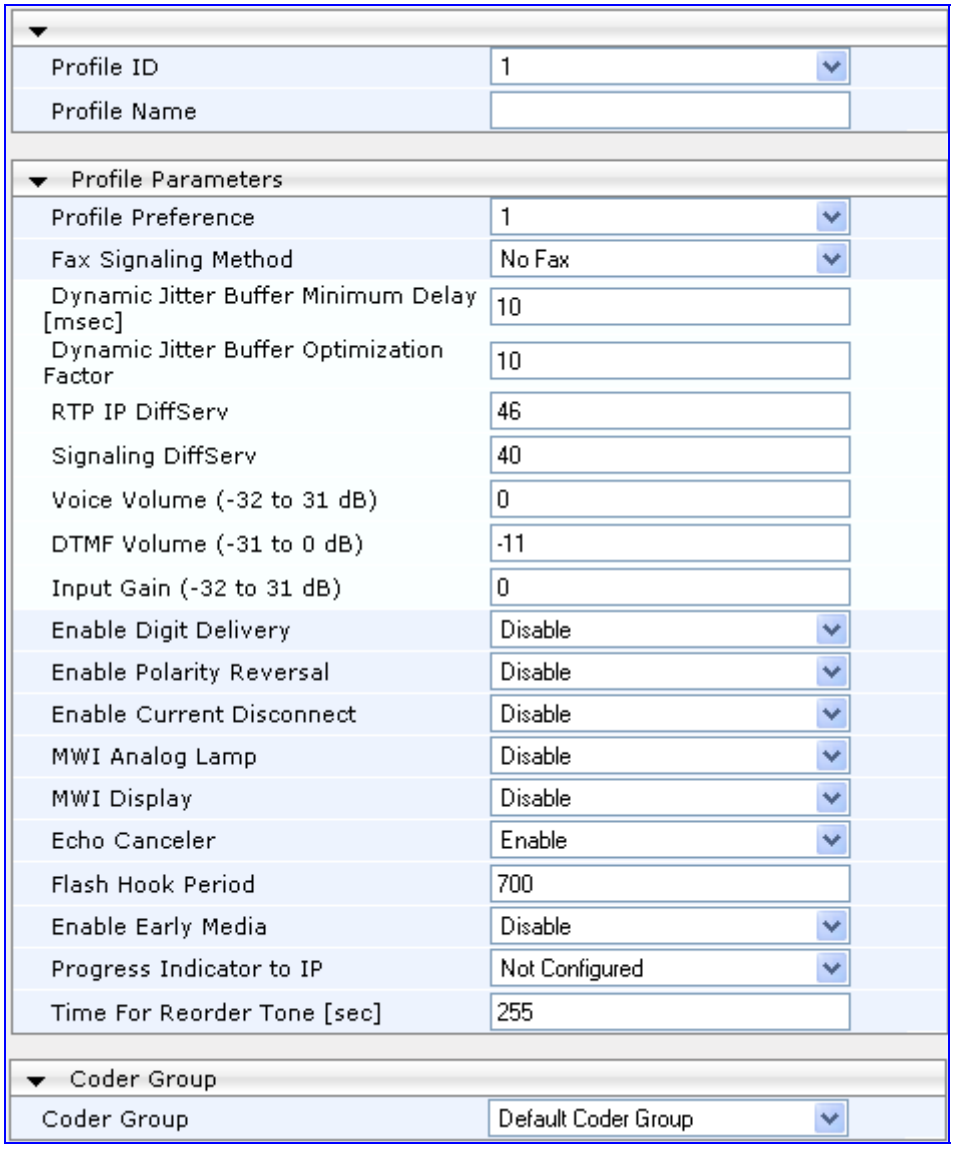
Notes:

- The following parameters are applicable only to Mediant 1000: 'Enable Polarity Reversal', 'Enable Current Disconnect', 'MWI Analog Lamp', and 'MWI Display'.
- You can also configure Tel Profiles using the *ini* file table parameter TelProfile (refer to "SIP Configuration Parameters" on page 323).

➢ To configure Tel Profiles, take these 9 steps:

1. Open the 'Tel Profile Settings' page (**Configuration** tab > **Protocol Configuration** menu > **Profile Definitions** submenu > **Tel Profile Settings** page item).

Figure 3-84: Tel Profile Settings Page



2. From the 'Profile ID' drop-down list, select the Tel Profile identification number you want to configure.

3. In the 'Profile Name' field, enter an arbitrary name that enables you to easily identify the Tel Profile.

4. From the 'Profile Preference' drop-down list, select the priority of the Tel Profile, where '1' is the lowest priority and '20' is the highest. If both IP and Tel profiles apply to the same call, the coders and other common parameters (noted by an asterisk in the description of the parameter TelProfile) of the preferred Profile are applied to that call. If the Preference of the Tel and IP Profiles is identical, the Tel Profile parameters are applied.
   Note: If the coder lists of both IP and Tel Profiles apply to the same call, only the coders common to both are used. The order of the coders is determined by the preference.

5. Configure the Profile's parameters according to your requirements. For detailed information on each parameter, refer to its description on the page in which it is configured as an individual parameter.
6. From the 'Coder Group' drop-down list, select the Coder Group (refer to "Coder Group Settings" on page 212) or the device's default coder (refer to "Coders" on page 157) to which you want to assign the Profile.
7. Repeat steps 2 through 6 to configure additional Tel Profiles (optional).
8. Click the **Submit** button to save your changes.
9. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

3.4.5.5.3 IP Profile Settings

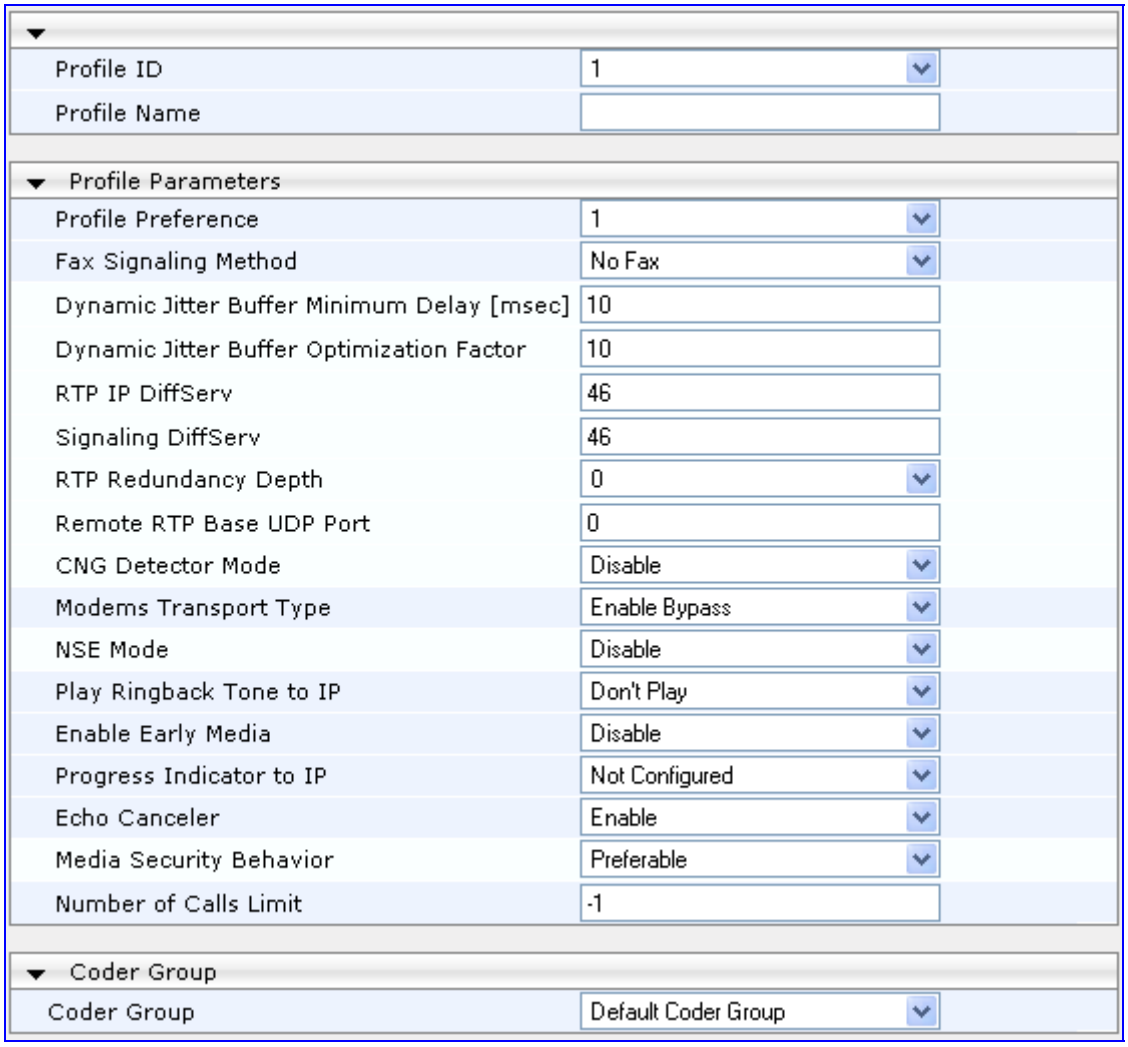
The 'IP Profile Settings' page allows you to define up to nine different IP Profiles. You can then assign these IP Profiles to routing rules in the 'Tel to IP Routing' page (refer to "Tel to IP Routing Table" on page 200) and 'IP to Trunk Group Routing' page (refer to "IP to Trunk Group Routing" on page 204). IP Profiles can also be used when working with a Proxy server (set AlwaysUseRouteTable to 1).

Note: You can also configure the IP Profiles using the *ini* file table parameter IPProfile (refer to "SIP Configuration Parameters" on page 323).

➢ To configure the IP Profile settings, take these 9 steps:

1. Open the 'IP Profile Settings' page (**Configuration** tab > **Protocol Configuration** menu > **Profile Definitions** submenu > **IP Profile Settings** page item).

Figure 3-85: IP Profile Settings Page



2. From the 'Profile ID' drop-down list, select an identification number for the IP Profile.
3. In the 'Profile Name' field, enter an arbitrary name that allows you to easily identify the IP Profile.
4. From the 'Profile Preference' drop-down list, select the priority of the IP Profile, where '1' is the lowest priority and '20' is the highest. If both IP and Tel profiles apply to the same call, the coders and other common parameters (noted by an asterisk in the description of the parameter IPProfile) of the preferred Profile are applied to that call. If the Preference of the Tel and IP Profiles is identical, the Tel Profile parameters are applied.
   Note: If the coder lists of both IP and Tel Profiles apply to the same call, only the coders common to both are used. The order of the coders is determined by the preference.

5. Configure the IP Profile's parameters according to your requirements. For detailed information on each parameter, refer to the description on the page in which it is configured as an individual parameter. Parameters that are unique to IP Profile are described in the table below.
6. From the 'Coder Group' drop-down list, select the coder group you want to assign to the Profile. You can select the device's default coders (refer to "Coders" on page 157) or one of the coder groups you defined in the 'Coder Group Settings' page (refer to "Coder Group Settings" on page 212).
7. Repeat steps 2 through 6 for the next IP Profiles (optional).
8. Click the **Submit** button to save your changes.
9. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-48: Description of Parameters Unique to IP Profile

Parameter	Description
Number of Calls Limit	Maximum number of concurrent calls. If the profile is set to some limit, the device maintains the number of concurrent calls (incoming and outgoing) pertaining to the specific profile. A limit value of '-1' indicates that there is no limitation on calls for that specific profile (default). A limit value of '0' indicates that all calls are rejected. When the number of concurrent calls is equal to the limit, the device rejects any new incoming and outgoing calls belonging to that profile.

3.4.5.6 Configuring the Endpoint Settings

The **Endpoint Settings** submenu allows you to configure port-specific parameters. This submenu includes the following page items:

- Authentication (refer to "Authentication" on page 219)
- Automatic Dialing (refer to "Automatic Dialing" on page 220)
- Caller Display Information (refer to "Caller Display Information" on page 221)
- Call Forward (refer to "Call Forward" on page 222)
- Caller ID Permissions (refer to "Caller ID Permissions" on page 224)
- Call Waiting (refer to "Call Waiting" on page 225)

Note: The Endpoint Settings menu is applicable only to Mediant 1000 analog modules.

3.4.5.6.1 Authentication

The 'Authentication' page defines a user name and password for authenticating each device port. Authentication is typically used for FXS interfaces, but can also be used for FXO interfaces.

Notes:

- The 'Authentication' page is applicable only to Mediant 1000.
- The 'Authentication Mode' parameter (refer to "Proxy & Registration Parameters" on page 146) determines whether authentication is performed per port or for the entire device. If authentication is performed for the entire device, the configuration in the 'Authentication' page is ignored.
- If either the user name or password field is omitted, the port's phone number (defined in Configuring the Trunk Group Table on page 226) and global password (refer to the parameter 'Password' described in "Proxy & Registration Parameters" on page 146) are used instead.
- You can also configure Authentication using the *ini* file table parameter Authentication (refer to "SIP Configuration Parameters" on page 323).

➢ To configure the Authentication Table, take these 5 steps:

1. Set the 'Authentication Mode' parameter to 'Per Endpoint' (refer to "Proxy & Registration Parameters" on page 146).
2. Open the 'Authentication' page (**Configuration** tab > **Protocol Configuration** menu > **Endpoint Settings** submenu > **Authentication** page item).

Figure 3-86: Authentication Screen

Gateway Port	User Name	Password
Module 3 Port 1 FXS	user1	1111
Module 3 Port 2 FXS	user2	2222
Module 4 Port 1 FXO		
Module 4 Port 2 FXO		
Module 4 Port 3 FXO		
Module 4 Port 4 FXO		

3. In the 'User Name' and 'Password' fields corresponding to a port, enter the user name and password respectively.
4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

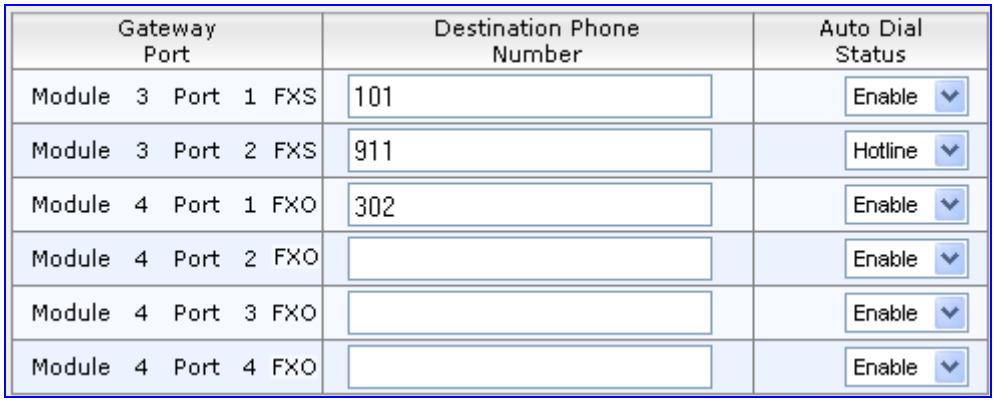
3.4.5.6.2 Automatic Dialing

The 'Automatic Dialing' page allows you to define a telephone number that is automatically dialed when an FXS or FXO port is used (e.g., off-hooked).

➢ To configure Automatic Dialing, take these 5 steps:

1. Open the 'Automatic Dialing' page (**Configuration** tab > **Protocol Configuration** menu > **Endpoint Settings** submenu > **Automatic Dialing** page item).

Figure 3-87: Automatic Dialing Page



Gateway Port	Destination Phone Number	Auto Dial Status
Module 3 Port 1 FXS	101	Enable
Module 3 Port 2 FXS	911	Hotline
Module 4 Port 1 FXO	302	Enable
Module 4 Port 2 FXO		Enable
Module 4 Port 3 FXO		Enable
Module 4 Port 4 FXO		Enable

2. In the 'Destination Phone Number' field corresponding to a port, enter the telephone number that you want automatically dialed.
3. From the 'Auto Dial Status' drop-down list, select one of the following:
   - Enable **[1]**: The number in the 'Destination Phone Number' field is automatically dialed if the phone is off-hooked (for FXS interfaces) or a ring signal (from PBX/PSTN switch) is generated to a port (FXO interfaces).
   - Disable **[0]**: The automatic dialing feature for the specific port is disabled (i.e., the number in the 'Destination Phone Number' field is ignored).
   - Hotline **[2]**: When a phone is off-hooked and no digit is dialed for a user-defined interval (Hotline Dial Tone Duration - refer to "DTMF & Dialing Parameters" on page 160), the number in the 'Destination Phone Number' field is automatically dialed (applies to FXS and FXO interfaces).
4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Notes:

- The 'Automatic Dialing' page is applicable only to Mediant 1000.
- After a ring signal is detected on an 'Enabled' FXO port, the device initiates a call to the destination number without seizing the line. The line is seized only after the call is answered.
- After a ring signal is detected on a 'Disabled' or 'Hotline' FXO port, the device seizes the line.
- You can also configure automatic dialing using the *ini* file table parameter TargetOfChannel (refer to "Analog Telephony Parameters" on page 355).

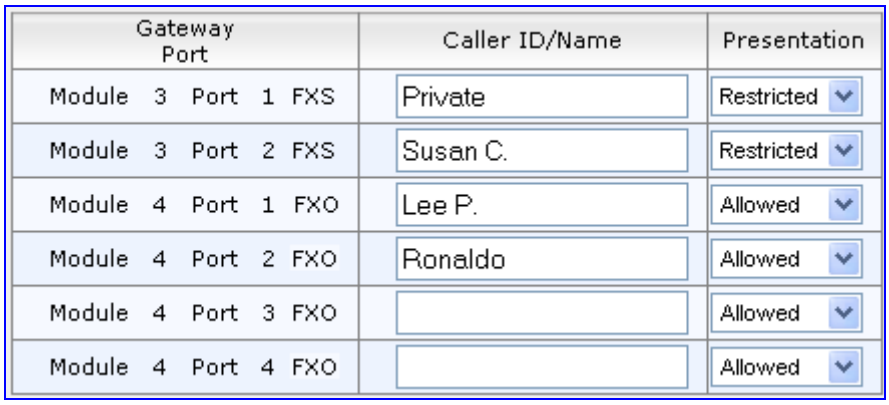
3.4.5.6.3 Caller Display Information

The 'Caller Display Information' page allows you to enable the device to send Caller ID information to IP when a call is made. The called party can use this information for caller identification. The information configured in this page is sent in an INVITE message in the 'From' header. For information on Caller ID restriction according to destination / source prefixes, refer to "Configuring the Number Manipulation Tables" on page 190.

➢ To configure the Caller Display Information, take these 5 steps:

1. Open the 'Caller Display Information' page (**Configuration** tab > **Protocol Configuration** menu > **Endpoint Settings** submenu > **Caller Display Information** page item).

Figure 3-88: Caller Display Information Page



Gateway Port	Caller ID/Name	Presentation
Module 3 Port 1 FXS	Private	Restricted
Module 3 Port 2 FXS	Susan C.	Restricted
Module 4 Port 1 FXO	Lee P.	Allowed
Module 4 Port 2 FXO	Ronaldo	Allowed
Module 4 Port 3 FXO		Allowed
Module 4 Port 4 FXO		Allowed

2. In the' Caller ID/Name' field corresponding to the desired port, enter the Caller ID string (up to 18 characters). Note that when the FXS ports receive 'Private' or 'Anonymous' strings in the From header, it doesn't send the calling name or number to the Caller ID display.
3. From the 'Presentation' drop-down list, select one of the following:
   - 'Allowed' **[0]** - sends the string defined in the 'Caller ID/Name' field when a Tel-to-IP call is made using the corresponding device port.
   - 'Restricted' **[1]** - the string defined in the 'Caller ID/Name' field is not sent (refer to the note below).

4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Notes:

- The 'Caller Display Information' page is relevant only for FXS and FXO interfaces and applicable only to Mediant 1000.
- If Caller ID name is detected on an FXO line (EnableCallerID = 1), it is used instead of the Caller ID name defined on this page.
- When the 'Presentation' field is set to 'Restricted', the Caller ID is sent to the remote side using only the P-Asserted-Identity and P-Preferred-Identity headers (AssertedIdMode).
- The value of the 'Presentation' field can be overridden by configuring the 'Presentation' field in the 'Source Number Manipulation' table (refer to "Configuring the Number Manipulation Tables" on page 190).
- You can also configure the Caller Display Information table using the *ini* file table parameter CallerDisplayInfo (refer to "Analog Telephony Parameters" on page 355).

3.4.5.6.4 Call Forward

The 'Call Forwarding Table' page allows you to forward (redirect) IP-to-Tel calls (using SIP 302 response) originally destined to specific device ports, to other device ports or to an IP destination.

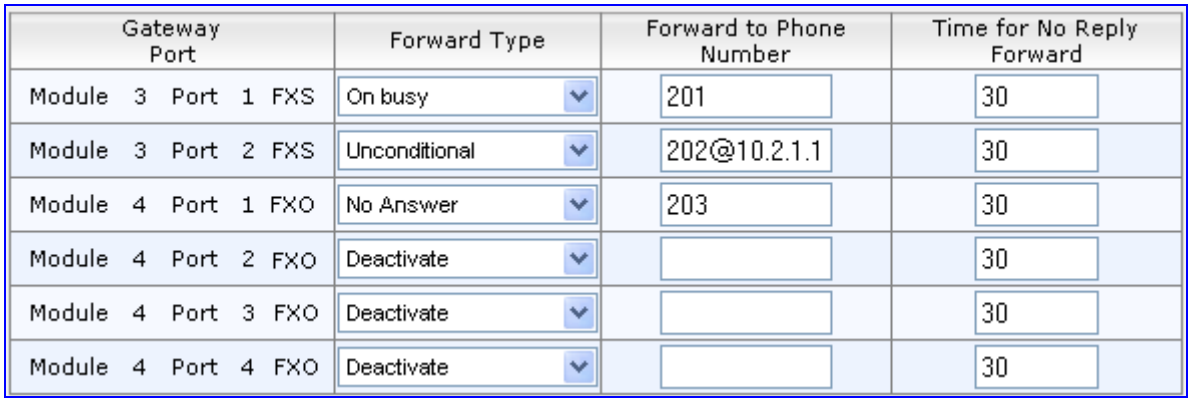
Notes:

- The 'Call Forwarding Table' page is relevant to FXS and FXO interfaces. In addition, it is applicable only to Mediant 1000.
- Ensure that the Call Forward feature is enabled (default) for the settings on this page to take effect. To enable Call Forward, use the parameter EnableForward ("Supplementary Services" on page 176).
- You can also configure the Call Forward table using the *ini* file table parameter FwdInfo (refer to "Analog Telephony Parameters" on page 355).

➢ To configure Call Forward per port, take these 4 steps:

1. Open the 'Call Forward Table' page (**Configuration** tab > **Protocol Configuration** menu > **Endpoint Settings** submenu > **Call Forward** page item).

Figure 3-89: Call Forward Table Page



Gateway Port	Forward Type	Forward to Phone Number	Time for No Reply Forward
Module 3 Port 1 FXS	On busy	201	30
Module 3 Port 2 FXS	Unconditional	202@10.2.1.1	30
Module 4 Port 1 FXO	No Answer	203	30
Module 4 Port 2 FXO	Deactivate		30
Module 4 Port 3 FXO	Deactivate		30
Module 4 Port 4 FXO	Deactivate		30

2. Configure the Call Forward parameters for each port according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-49: Call Forward Table

Parameter	Description
Forward Type	Determines the scenario for forwarding a call. ▪ **[0]** Deactivate = Don't forward incoming calls (default). ▪ **[1]** On Busy = Forward incoming calls when the port is busy. ▪ **[2]** Unconditional = Always forward incoming calls. ▪ **[3]** No Answer = Forward incoming calls that are not answered within the time specified in the 'Time for No Reply Forward' field. ▪ **[4]** On Busy or No Answer = Forward incoming calls when the port is busy or when calls are not answered within the time specified in the 'Time for No Reply Forward' field. ▪ **[5]** Do Not Disturb = Immediately reject incoming calls.
Forward to Phone Number	The telephone number or URI (<number>@<IP address>) to where the call is forwarded. **Note:** If this field only contains a telephone number and a Proxy isn't used, the 'forward to' phone number must be specified in the 'Tel to IP Routing' table (refer to "Tel to IP Routing Table" on page 200).
Time for No Reply Forward	If you have set the 'Forward Type' for this port to 'No Answer', enter the number of seconds the device waits before forwarding the call to the phone number specified.

3.4.5.6.5 Caller ID Permissions

The 'Caller ID Permissions' page allows you to enable or disable (per port), the Caller ID generation (for FXS interfaces) and detection (for FXO interfaces). If a port isn't configured, its Caller ID generation / detection are determined according to the global parameter EnableCallerID described in "Supplementary Services" on page 176.

Notes:

- The 'Caller ID Permissions' page is relevant to FXS and FXO interfaces and applicable only to Mediant 1000.
- You can also configure the Caller ID Permissions table using the *ini* file table parameter EnableCallerID (refer to "Analog Telephony Parameters" on page 355).

➢ To configure Caller ID Permissions per port, take these 4 steps:

1. Open the 'Caller ID Permissions' page (**Configuration** tab > **Protocol Configuration** menu > **Endpoint Settings** submenu > **Caller ID Permissions** page item).

Figure 3-90: Caller ID Permissions Page

Gateway Port	Caller ID
Module 3 Port 1 FXS	Enable
Module 3 Port 2 FXS	Disable
Module 4 Port 1 FXO	
Module 4 Port 2 FXO	
Module 4 Port 3 FXO	
Module 4 Port 4 FXO	

2. From the 'Caller ID' drop-down list, select one of the following:
   - 'Enable': Enables Caller ID generation (FXS) or detection (FXO) for the specific port.
   - 'Disable': Caller ID generation (FXS) or detection (FXO) for the specific port is disabled.
   - Not defined: Caller ID generation (FXS) or detection (FXO) for the specific port is determined according to the parameter 'Enable Caller ID' (described in "Supplementary Services" on page 176).
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

3.4.5.6.6 Call Waiting

The 'Call Waiting' page allows you to enable or disable call waiting per device FXS port.

Notes:

- This page is applicable only to FXS interfaces and applicable only to Mediant 1000.
- Instead of using this page, you can enable or disable call waiting for all the device's ports, using the global call waiting parameter 'Enable Call Waiting' (refer to "Supplementary Services" on page 176).
- You can also configure the Call Waiting table using the *ini* file table parameter CallWaitingPerPort (refer to "SIP Configuration Parameters" on page 323).

➢ To configure Call Waiting, take these 4 steps:

1. Open the 'Caller Waiting' page (**Configuration** tab > **Protocol Configuration** menu > **Endpoint Settings** submenu > **Call Waiting** page item).

Figure 3-91: Caller Waiting Page

Gateway Port	Call Waiting Configuration
Module 3 Port 1 FXS	Enable
Module 3 Port 2 FXS	Enable
Module 4 Port 1 FXO	
Module 4 Port 2 FXO	
Module 4 Port 3 FXO	
Module 4 Port 4 FXO	

2. From the 'Call Waiting Configuration' drop-down list corresponding to the port you want to configure for call waiting, select one of the following options:
   - 'Enable': Enables call waiting for the specific port. When the device receives a call on a busy endpoint (port), it responds with a 182 response (and not with a 486 busy). The device plays a call waiting indication signal. When hook-flash is detected by the device, the device switches to the waiting call. The device that initiated the waiting call plays a Call Waiting Ringback tone to the calling party after a 182 response is received.
   - 'Disable': No call waiting for the specific port.
   - Empty: Call waiting is determined according to the global parameter 'Enable Call Waiting' (described in "Supplementary Services" on page 176).
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

3.4.5.7 Configuring the Trunk and IP Groups

The Trunk/**IP Group** menu allows you to configure groups of channels. This submenu includes the following page items:

- Trunk Group (refer to "Configuring the Trunk Group Table" on page 226)
- Trunk Group Settings (refer to "Configuring the Trunk Group Settings" on page 228)
- IP Group Table (refer to "Configuring the IP Groups" on page 231)
- Account Table (refer to "Configuring the Account Table" on page 232)

3.4.5.7.1 Configuring the Trunk Group Table

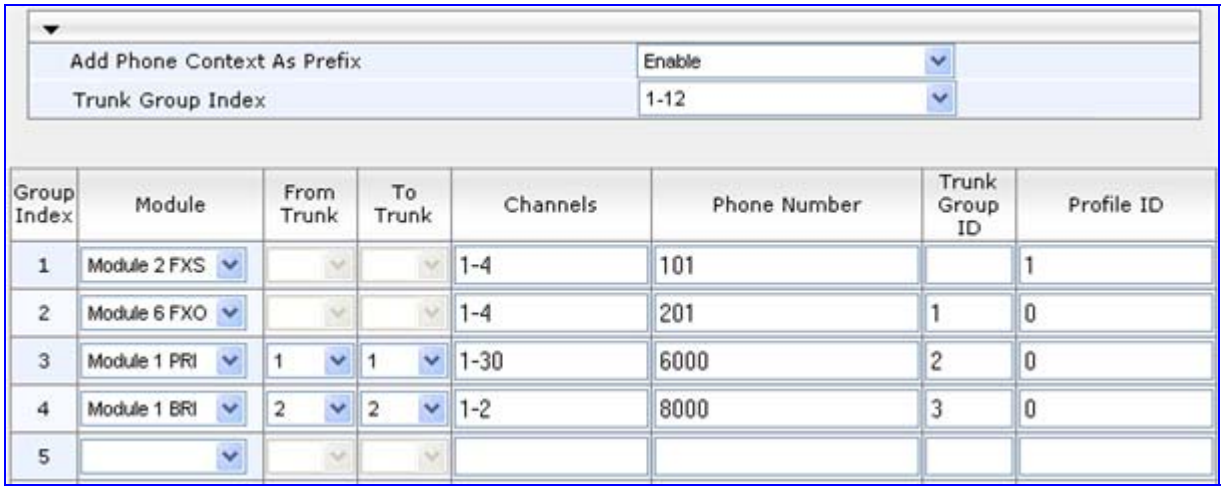
The 'Trunk Group Table' page provides you with a table for enabling device channels, by assigning them telephone numbers, Trunk Groups, and Profiles. Trunk Groups are used for routing IP-to-Tel calls with common rules. Channels that are not defined are disabled. You can add up to 24entries in this table.

Note: You can also configure the Trunk Groups using the *ini* file table parameter TrunkGroup_x to (refer to "Number Manipulation and Routing Parameters" on page 364).

➢ **To configure the Trunk Group table, take these 4 steps:**

1. Open the 'Trunk Group Table' page (**Configuration** tab > **Protocol Configuration** menu > **Trunk/IP Group** submenu > **Trunk Group** page item).

Figure 3-92: Trunk Group Table Page



2. Configure the Trunk Group according to the table below**.**
3. Click the **Submit** button to save your changes.
4. To save the changes to the flash memory, refer to "Saving Configuration" on page 265.

Table 3-50: Trunk Group Table Description

Parameter	Description
Module	The module type (FXS, FXO, BRI, or PRI) for which you want to define the Trunk Group. **Note:** FXS and FXO modules are applicable only to Mediant 1000.
From Trunk	Starting physical Trunk number. The number of listed Trunks depends on the device's hardware configuration. **Note:** Applicable only to PRI and BRI modules.
To Trunk	Ending physical Trunk number. The number of listed Trunks depends on the device's hardware configuration. **Note:** Applicable only to PRI and BRI modules.
Channels	The device's channels or ports (analog module), or Trunk B-channels (digital module). To enable the channels, enter the channel numbers. You can enter a range of channels by using the format [n-m], where *n* represents the lower channel number and *m* the higher channel number, e.g., [1-24] specifies channels 1 through 24. **Notes:** ▪ The number of defined channels must not exceed the number of the Trunk's B-channels. ▪ To represent all channels, enter a single asterisk (*).
Phone Number	Enter the first telephone number that you want to assign to the first channel defined in the 'Channels' field. Subsequent channels are assigned the next consecutive phone number. **Note:** This field is optional. The logical numbers defined in this field are used when an incoming PSTN / PBX call doesn't contain the calling number or called number (the latter being determined by the parameter ReplaceEmptyDstWithPortNumber); these numbers are used to replace them. These logical numbers are also used for channel allocation for IP-to-Tel calls if the Trunk Group's 'Channel Select Mode' is set to 'By Dest Phone Number'.
Trunk Group ID	The Trunk Group ID (1-99) assigned to the corresponding channels. The same Trunk Group ID can be used for more than one group of channels. Trunk Group ID is used to define a group of common channel behavior that are used for routing IP-to-Tel calls. If an IP-to-Tel call is assigned to a Trunk Group, the call is routed to the channel or channels that correspond to the Trunk Group ID. You can configure the 'Trunk Group Settings' table (refer to "Configuring the Trunk Group Settings" on page 228) to determine the method in which new calls are assigned to channels within the Trunk Groups. **Note:** You must configure the 'IP to Trunk Group Routing Table' page (refer to "IP to Trunk Group Routing" on page 204) to assign incoming IP calls to the appropriate Trunk Group. If you do not configure the 'IP to Trunk Group Routing Table', calls do not complete.
Profile ID	The Tel profile ID (refer to "Tel Profile Settings" on page 214) assigned to the channels defined in the 'Channels' field.

3.4.5.7.2 Configuring the Trunk Group Settings

The 'Trunk Group Settings' page is mainly used to select the method for which IP-to-Tel calls are assigned to channels within each Trunk Group. If no method is selected (for a specific Trunk Group), the setting of the global parameter ChannelSelectMode in the 'SIP General Parameters' page (refer to "SIP General Parameters" on page 134) applies. In addition, this page also defines the method for registering Trunk Groups to selected Serving IP Group IDs (if defined). You can add up to 24entries in this table.

Note: You can also configure the Trunk Group Settings table using the *ini* file table parameter TrunkGroupSettings (refer to "Number Manipulation and Routing Parameters" on page 364).

➢ **To configure the Trunk Group Settings table, take these 5 steps:**

1. Open the 'Trunk Group Settings' page (**Configuration** tab > **Protocol Configuration** menu > Trunk/**IP Group** submenu > Trunk **Group Settings** page item).

Figure 3-93: Trunk Group Settings Page

2. From the 'Routing Index' drop-down list, select the range of entries that you want to edit (up to 24 entries can be configured).
3. Configure the Trunk Group according to the table below.
4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-51: Trunk Group Settings Parameters Description

Mode	Description
Trunk **Group ID**	The Trunk Group ID that you want to configure. The valid range is 1-99. Trunks are assigned to Trunk Groups in the 'Trunk Group Table' page (refer to "Configuring the Trunk Group Table" on page 226).

Mode	Description
Channel Select Mode	The method in which IP-to-Tel calls are assigned to channels pertaining to a Trunk Group: ▪ By Dest Phone Number = Selects the device's channel according to the called number defined in the 'Trunk Group Table' (refer to "Configuring the Trunk Group Table" on page 226). ▪ Cyclic Ascending (default) = Selects the next available channel in an ascending cyclic order. The next highest channel number in the Trunk Group is always selected. When the highest channel number in the Trunk Group is reached, the lowest channel number in the Trunk Group is selected, and then it starts ascending again. ▪ Ascending = Selects the lowest available channel. The lowest channel number in the Trunk Group is always first selected, and if that channel is unavailable, the next highest channel is selected. ▪ Cyclic Descending = Selects the next available channel in descending cyclic order. The next lowest channel number in the Trunk Group is always first selected. When the lowest channel number in the Trunk Group is reached, it selects the highest channel number in the Trunk Group and then start descending again. ▪ Descending = Selects the highest available channel. The highest channel number in the Trunk Group is always first selected, and if that channel is unavailable, the next lowest channel is selected. ▪ Dest Number + Cyclic Ascending = The channel is first selected according to the called number. If the called number isn't found, the next available channel in ascending cyclic order is selected. Note that if the called number is found, but the channel associated with the number is busy, the call is released. ▪ By Source Phone Number = Selects the channel according to the calling number. ▪ Trunk Cyclic Ascending = The first channel of the next Trunk (i.e., next to the Trunk from which the previous channel was allocated) is selected. **Note:** Not applicable for analog interfaces.
Registration Mode	Registration mode per Trunk Group: ▪ Per Gateway = Single registration for the entire device (default). This mode is applicable only if a default Proxy or Registrar IP are configured, and Registration is enabled (i.e., parameter IsRegisterUsed is set to 1). In this mode, the URI userpart in the From, To, and Contact headers is set to the value of the global registration parameter GWRegistrationName (refer to "Proxy & Registration Parameters" on page 146) or username if GWRegistrationName is not configured. ▪ Per Endpoint = Each channel in the Trunk Group registers individually. The registrations are sent to the ServingIPGroupID if defined in the table, otherwise to the default Proxy, and if no default Proxy, then to the Registrar IP. ▪ Don't Register = No registrations are sent by endpoints pertaining to the Trunk Group. For example, if the device is configured globally to register all its endpoints (using the parameter ChannelSelectMode), you can exclude some endpoints from being registered by assigning them to a Trunk Group and configuring the Trunk Group registration mode to 'Don't Register'. ▪ Per Account = Registrations are sent (or not) according to the settings in the Account table (refer to "Configuring the Account Table" on page 232). **Notes:**

Mode	Description
	▪ To enable Trunk Group registrations, configure the global parameter IsRegisterNeeded to 1. This is unnecessary for 'Per Account' registration mode. ▪ If no mode is selected, the registration is performed according to the global registration parameter ChannelSelectMode (refer to "Proxy & Registration Parameters" on page 146). ▪ If the device is configured globally (ChannelSelectMode) to register Per Endpoint, and a Trunk Group comprising four channels is configured to register Per Gateway, the device registers all channels except the first four channels. The Trunk Group of these four channels sends a single registration request.
Serving IP Group ID	The Serving IP Group ID to where INVITE messages initiated by the Trunk Group endpoints are sent. The actual destination to where these INVITE messages are sent is to the Proxy Set ID (refer to "Proxy Sets Table" on page 153) associated with this Serving IP Group. The Request URI hostname in the INVITE and REGISTER messages (except for 'Per Account' registration modes) is set to the value for the field 'SIP Group Name' defined in the 'IP Group' table (refer to "Configuring the IP Groups" on page 231). If no ServingIPGroupID is selected, the INVITE messages are sent to the default Proxy or according to the 'Tel to IP Routing Table' (refer to "Tel to IP Routing Table" on page 200). **Note:** If the parameter PreferRouteTable is set to 1 (refer to "Proxy & Registration Parameters" on page 146), the routing rules in the 'Tel to IP Routing Table' prevails over the selected ServingIPGroupID.
Gateway Name	The host name used in the From header in INVITE messages, and as a host name in From/To headers in REGISTER requests. If not configured, the global parameter SIPGatewayName is used instead.
Contact User	This is used as the user part in the Contact URI in INVITE messages, and as a user part in From, To, and Contact headers in REGISTER requests. This is applicable only if the field 'Registration Mode' is set to 'Per Account', and the Registration through the Account table is successful. **Notes:** • If registration fails, then the userpart in the INVITE Contact header contains the source party number. • The 'ContactUser' parameter in the 'Account Table' page overrides this parameter.

An example is shown below of a REGISTER message for registering endpoint "101" using registration Per Endpoint mode. The "SipGroupName" in the request URI is taken from the IP Group table.

```
REGISTER sip:SipGroupName SIP/2.0
Via: SIP/2.0/UDP 10.33.37.78;branch=z9hG4bKac862428454
From: <sip:101@GatewayName>;tag=1c862422082
To: <sip:101@GatewayName>
Call-ID: 990797706251200023282 5@10.33.37.78
CSeq: 3 REGISTER
Contact: <sip:101@10.33.37.78>;expires=3600
Expires: 3600
User-Agent: Audiocodes-Sip-Gateway/v.5.40A.008.002
Content-Length: 0
```

3.4.5.7.3 Configuring the IP Groups

The 'IP Group Table' page allows you to create logical IP entities (IP Groups) that are later used in the call routing tables.

The IP Groups are typically implemented in Tel-to-IP call routing. The IP Group can be used as a destination entity (DestIPGroupID) in the 'Tel to IP Routing' table, and ServingIPGroupID in the 'Trunk Group Settings' (refer to "Configuring the Trunk Group Settings" on page 228) and 'Account' (refer to "Configuring the Account Table" on page 232) tables. These call routing tables are used for identifying the IP Group from where the INVITE is sent for obtaining a digest user/password from the 'Account' table if there is a need to authenticate subsequent SIP requests in the call. The IP Group can also be implemented in IP-to-Tel call routing as a source IP Group (SourceIPGroupID).

The IP Groups are assigned various entities such as a Proxy Set ID, which represents an IP address (created in "Proxy Sets Table" on page 153). You can also assign the IP Group with a host name and other parameters that reflect parameters sent in SIP Request From\To.

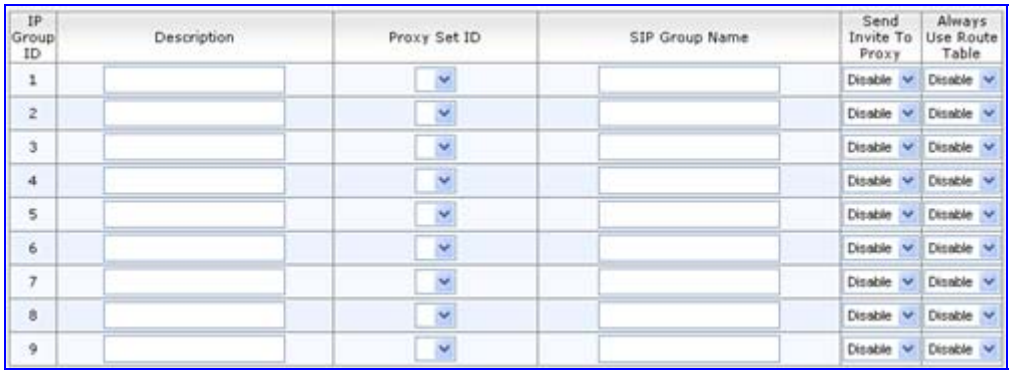
Notes:

- By default, if you disable the use of a proxy (i.e., IsProxyUsed is set to 0), then only one IP Group is defined (and working with multiple IP Groups is not valid).
- You can also configure the IP Groups table using the *ini* file table parameter IPGroup (refer to "SIP Configuration Parameters" on page 323).

➢ To configure IP Groups, take these 4 steps:

1. Open the 'IP Group Table' page (**Configuration** tab > **Protocol Configuration** menu > Trunk/**IP Group** submenu > **IP Group Table** page item).

Figure 3-94: IP Group Table Page



2. Configure the IP group parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-52: IP Group Parameters Description

Parameter	Description
IP Group ID	The identifying number of the IP Group. The range is 1-9.
Description	Brief string description of the IP Group. The value range is a string of up to 29 characters. The default is an empty field.
Proxy Set ID	Selects the Proxy Set ID to be associated with the IP Group. All INVITE messages configured to be 'sent' to the specific IP Group are physically sent to the IP address associated with the Proxy Set. The range is 0-5, where 0 is the default Proxy Set. **Note:** To configure Proxy Set ID's, refer to "Proxy Sets Table" on page 153.
SIP Group Name	The request URI host name used in INVITE and REGISTER messages. If not specified, the value of the global parameter ProxyName (refer to "Proxy & Registration Parameters" on page 146) is used instead. The value range is a string of up to 49 characters. The default is an empty field.
Send Invite To Proxy	Determines whether INVITE messages are sent to the IP Group. ▪ Disable (default). ▪ Enable = INVITE messages as a result of REFER/3xx messages are sent to the IP Group, overriding the Refer-To/Contact destination.
Always Use Route Table	Determines the Request URI host name in outgoing INVITE messages. ▪ Disable (default). ▪ Enable = The device uses the IP address (or domain name) defined in the 'Tel to IP Routing' table ("Tel to IP Routing Table" on page 200) as the Request URI host name in outgoing INVITE messages, instead of the value entered in the 'SIP Group Name' field.

3.4.5.7.4 Configuring the Account Table

The 'Account Table' page allows you to define *accounts* per Trunk Groups (referred to as *Served Trunk Group*) for registration and/or digest authentication (user name and password) to a destination IP address (Serving IP Group). The Accounts table can be used, for example, to register to an Internet Telephony Service Provider (ITSP) on behalf of an IP-PBX to which the device is connected. The registrations are sent to the Proxy Set ID (refer to "Proxy Sets Table" on page 153) associated with these Serving IP Groups. A Trunk Group can register to more than one Serving IP Group (e.g., ITSP's), by configuring multiple entries in this Account table with the same Served Trunk Group, but with different Serving IP Groups, user name/password, Host Name, and Contact User parameters.

Note: You can also configure the Account table using the *ini* file table parameter Account (refer to "SIP Configuration Parameters" on page 323).

➢ To configure Accounts, take these 5 steps:

1. Open the 'Account Table' page (**Configuration** tab > **Protocol Configuration** menu > Trunk/**IP Group** submenu > **Account Table** page item).

Figure 3-95: Account Table Page

2. To add an Account, in the 'Add' field, enter the desired table row index, and then click **Add**. A new row appears.
3. Configure the Account parameters according to the table below.
4. Click the **Apply** button to save your changes.
5. To save the changes, refer to "Saving Configuration" on page 265.

Note: For a description of the web interface's table command buttons (e.g., **Duplicate** and **Delete**), refer to "Working with Tables" on page 36.

Table 3-53: Account Parameters Description

Parameter	Description
Served Trunk Group	The Trunk Group ID for which the device performs registration and/or authentication to a destination IP Group (i.e., Serving IP Group). For Tel-to-IP calls, the ServedTrunkGroup is the source Trunk Group from where the call initiated. For IP-to-Tel calls, the ServedTrunkGroup is the 'Trunk Group ID' defined in the 'IP to Trunk Group Routing' table (refer to "IP to Trunk Group Routing" on page 204). For defining Trunk Groups, refer to "Configuring the Trunk Group Table" on page 226.
Serving IP Group	The destination Serving IP Group ID (defined in "Configuring the IP Groups" on page 231) to where the REGISTER requests (if enabled) are sent. The actual destination to where the Trunk Group registers is the IP address defined for the Proxy Set ID (refer to "Proxy Sets Table" on page 153) associated with the Serving IP Group. This occurs only in the following conditions: ▪ The parameter 'Registration Mode' is set to 'Per Account' in the 'Trunk Group Settings' table (refer to "Configuring the Trunk Group Settings" on page 228). ▪ The parameter 'Register' in this table is set to 1. In addition, for a SIP call that is identified by both the ServedTrunkGroup and ServingIPGroup, the username and password for digest authentication defined in this table is used. For Tel-to-IP calls, the ServingIPGroup is the destination IP Group defined in the 'Trunk Group Settings' table or 'Tel to IP Routing' table (refer to "Tel to IP Routing Table" on page 200). For IP-to-Tel calls, the ServingIPGroup is the 'Source IPGroup ID' defined in the 'IP to Trunk Group Routing' table (refer to "IP to Trunk Group Routing" on page 204). **Note:** If no match is found in the 'Account' table for incoming or outgoing calls,

Parameter	Description
	the username and password defined in the 'Authentication' table for FXS interfaces (refer to Authentication on page 219) or the global parameters (UserName and Password) defined on the 'Proxy & Registration' page (refer to "Proxy & Registration Parameters" on page 146) are used.
Username	Digest MD5 Authentication user name (up to 50 characters).
Password	Digest MD5 Authentication password (up to 50 characters).
HostName	Defines the Address of Record (AOR) host name. It appears in REGISTER From/To headers as ContactUser@HostName. For successful registrations, this HostName is also included in the INVITE request's From header URI. If not configured or if registration fails, the 'Gateway Name' parameter from the 'Trunk Group Settings' page (refer to "Configuring the Trunk Group Settings" on page 228) is used instead. This parameter can be up to 49 characters.

Register	Enables registration.
	▪ **0** = Disable.
	▪ **1** = Enable.
	When enabled, the device sends REGISTER requests to the ServingIPGroup. In addition, to activate the 'Account' table registration, you also need to set the parameter 'Registration Mode' to 'Per Account' in the 'Trunk Group Settings' table (refer to "Configuring the Trunk Group Settings" on page 228) for the specific Trunk Group. The HostName (Host name in From/To headers) and ContactUser (user in From/To and Contact headers) are taken from this 'Account' table upon a successful registration. See the example below:
	`REGISTER sip:audiocodes SIP/2.0`
	`Via: SIP/2.0/UDP 10.33.37.78;branch=z9hG4bKac1397582418`
	`From: <sip:ContactUser@HostName>;tag=1c1397576231`
	`To: <sip: ContactUser@HostName >`
	`Call-ID: 139756895726120002256@10.33.37.78`
	`CSeq: 1 REGISTER`
	`Contact: <sip:ContactUser@10.33.37.78>;expires=3600`
	`Expires: 3600`
	`User-Agent: Audiocodes-Sip-Gateway/v.5.40A.008.002`
	`Content-Length: 0`
	Note: The Trunk Group account registration is not effected by the parameter IsRegisterNeeded.

Contact User	Defines the AOR user name. It appears in REGISTER From/To headers as ContactUser@HostName, and in INVITE/200 OK Contact headers as ContactUser@<device's IP address>. If not configured, the 'Contact User' parameter from the 'Trunk Group Settings' page (refer to "Configuring the Trunk Group Settings" on page 228) is used instead. **Note:** If registration fails, then the userpart in the INVITE Contact header contains the source party number.

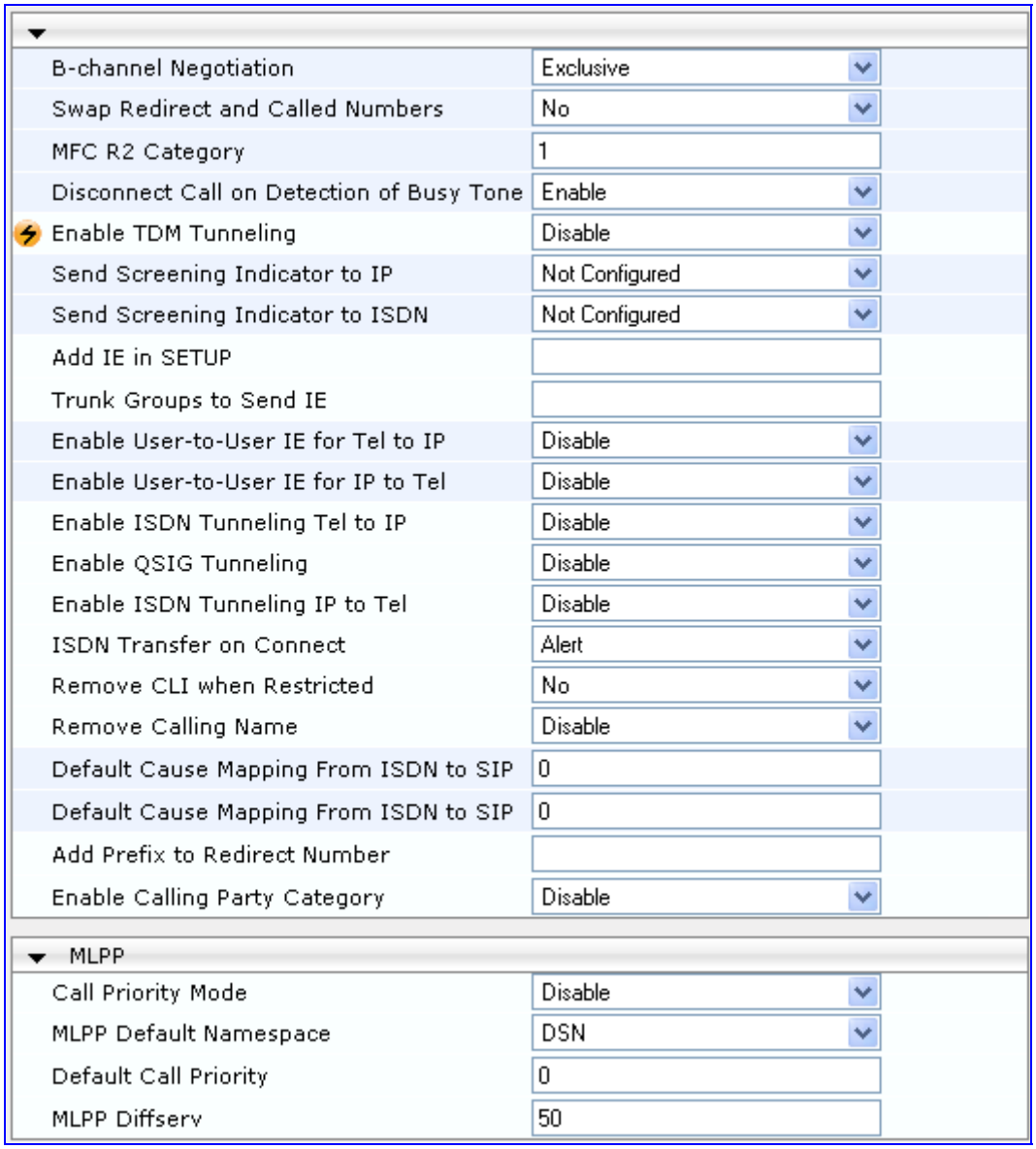
3.4.5.8 Configuring the Digital Gateway Parameters

The 'Digital Gateway Parameters' page allows you to configure miscellaneous digital parameters.

➢ To configure the digital gateway parameters, take these 4 steps:

1. Open the 'Digital Gateway Parameters' page (**Configuration** tab > **Protocol Configuration** menu > **Digital Gateway** submenu > **Digital Gateway Parameters** page item).

Figure 3-96: Digital Gateway Parameters Page



2. Configure the Digital Gateway parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-54: Digital Gateway Parameters Description

Parameter	Description
B-channel Negotiation **[BchannelNegotiation]**	Determines the ISDN B-Channel negotiation mode. ▪ **[0]** Preferred ▪ **[1]** Exclusive (default) ▪ **[2]** Any **Notes:** ▪ Applicable only to ISDN protocols. ▪ For some ISDN variants, when 'Any' (2) is selected, the SETUP message does not include the Channel Identification IE. ▪ The 'Any' (2) option is applicable only if the parameter 'ISDN Termination Side' is set to 'Use side' (refer to "Configuring the Trunk Settings" on page 96).
Swap Redirect and Called Numbers **[SwapRedirectNumber]**	▪ **[0]** No = Don't change numbers (default). ▪ **[1]** Yes = Incoming ISDN call that includes a redirect number (sometimes referred to as 'original called number') uses the redirect number instead of the called number.
MFC R2 Category **[R2Category]**	Determines the tone for MFC R2 Calling Party Category (CPC). The parameter provides information on the calling party such as National or International call, Operator or Subscriber and Subscriber priority. The value range is 1 to 15 (defining one of the MFC R2 tones). The default value is 1.
Disconnect Call on Detection of Busy Tone **[DisconnectOnBusyTone]**	Determines whether a call is disconnected upon detection of a busy tone. ▪ **[0]** Disable = Do not disconnect call on detection of busy tone. ▪ **[1]** Enable = Disconnect call on detection of busy tone (default). **Note:** This parameter is applicable only to CAS protocols. For ISDN, use the parameter ISDNDisconnectOnBusyTone.
Enable TDM Tunneling **[EnableTDMoverIP]**	Enables TDM tunneling. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = TDM Tunneling is enabled. When TDM Tunneling is enabled, the originating device automatically initiates SIP calls from all enabled B-channels pertaining to E1/T1/J1 spans that are configured with the 'Transparent' protocol. The called number of each call is the internal phone number of the B-channel from where the call originates. The 'IP to Trunk Group' routing table is used to define the destination IP address of the terminating device. The terminating device automatically answers these calls if its E1/T1 protocol is set to 'Transparent' (ProtocolType = 5).

Parameter	Description
Send Screening Indicator to IP **[ScreeningInd2IP]**	Overrides the calling party's number (CPN) screening indication in the received ISDN SETUP message for Tel-to-IP calls. ▪ **[-1]** Not Configured = not configured (interworking from ISDN to IP) or set to 0 for CAS (default). ▪ **[0]** User Provided = CPN set by user, but not screened (verified). ▪ **[1]** User Passed = CPN set by user, verified and passed. ▪ **[2]** User Failed = CPN set by user, and verification failed. ▪ **[3]** Network Provided = CPN set by network. **Note:** Applicable only if Remote Party ID (RPID) header is enabled.
Send Screening Indicator to ISDN **[ScreeningInd2ISDN]**	Overrides the screening indicator of the calling party's number for IP-to-Tel ISDN calls. ▪ **[-1]** Not Configured = Not configured (interworking from IP to ISDN) (default). ▪ **[0]** User Provided = user provided, not screened. ▪ **[1]** User Passed = user provided, verified and passed. ▪ **[2]** User Failed = user provided, verified and failed. ▪ **[3]** Network Provided = network provided.
Add IE in SETUP **[AddIEinSetup]**	Adds an optional Information Element (IE) data (in hex format) to ISDN SETUP messages. For example, to add IE '0x20,0x02,0x00,0xe1', enter the following value for this parameter: '200200e1'. **Note:** This IE is sent from the Trunk Group IDs defined by the parameter SendIEonTG.
Trunk Groups to Send IE **[SendIEonTG]**	Defines Trunk Group IDs (up to 50 characters) from where the optional ISDN IE defined by the parameter AddIEinSetup is sent. For example: '1,2,4,10,12,6'.
Enable User-to-User IE for Tel to IP **[EnableUUITel2IP]**	Enables ISDN PRI-to-SIP interworking. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enable transfer of User-to-User Information Element (UUIE) from PRI to SIP. The device supports the following ISDN PRI-to-SIP interworking: SETUP to SIP INVITE, CONNECT to SIP 200 OK, USER INFORMATION to SIP INFO, ALERT to SIP 18x response, and DISCONNECT to SIP BYE response messages. **Note:** The interworking of User-to-User IE to SIP INFO is supported only on the 4ESS PRI variant.

Parameter	Description
Enable User-to-User IE for IP to Tel **[EnableUUIIP2Tel]**	Enables SIP-to-PRI ISDN interworking. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enable transfer of UUIE from SIP INVITE message to PRI SETUP message. The device supports the following SIP-to-PRI ISDN interworking: SIP INVITE to SETUP, SIP 200 OK to CONNECT, SIP INFO to USER INFORMATION, SIP 18x to ALERT, and SIP BYE to DISCONNECT. **Note:** The interworking of User-to-User IE to SIP INFO is supported only on 4ESS PRI variant.
Enable ISDN Tunneling Tel to IP **[EnableISDNTunnelingTel2IP]**	Enables ISDN Tunneling. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Using Header = Enable ISDN Tunneling from ISDN PRI to SIP using a proprietary SIP header. ▪ **[2]** Using Body = Enable ISDN Tunneling from ISDN PRI to SIP using a dedicated message body. When ISDN Tunneling is enabled, the device sends all ISDN PRI messages using the correlated SIP messages. The ISDN SETUP message is tunneled using SIP INVITE, all mid-call messages are tunneled using SIP INFO, and ISDN DISCONNECT / RELEASE is tunneled using SIP BYE messages. The raw data from the ISDN is inserted into a proprietary SIP header (X-ISDNTunnelingInfo) or a dedicated message body (application/isdn) in the SIP messages. **Note:** It is necessary to set the parameter ISDNDuplicateQ931BuffMode to 128 (i.e., duplicate all messages) for this feature to function.
Enable QSIG Tunneling **[EnableQSIGTunneling]**	Enables QSIG tunneling over SIP according to <draft-elwell-sipping-qsig-tunnel-03>. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Enable = Enable QSIG tunneling from QSIG to SIP and vice versa. When QSIG tunneling is enabled, all QSIG messages are sent as raw data in corresponding SIP messages using a dedicated message body. **Notes:** ▪ QSIG tunneling must be enabled on both originating and terminating devices. ▪ To enable this function, set the parameter ISDNDuplicateQ931BuffMode to 128 (i.e., duplicate all messages).

Parameter	Description
Enable ISDN Tunneling IP to Tel **[EnableISDNTunnelingIP2Tel]**	▪ **[0]** Disable = Disable (default). ▪ **[1]** Using Header = Enable ISDN Tunneling from SIP to ISDN PRI using a proprietary SIP header. ▪ **[2]** Using Body = Enable ISDN Tunneling from SIP to ISDN PRI using a dedicated message body. When ISDN Tunneling is enabled, the device extracts raw data received in a proprietary SIP header (X-ISDNTunnelingInfo) or a dedicated message body (application/isdn) in the SIP messages and sends the data as ISDN messages to the PSTN side.
ISDN Transfer On Connect **[SendISDNTransferOnConnect]**	This parameter is used for the ECT/TBCT/RLT/Path Replacement ISDN Transfer methods. Usually, the device requests the PBX to connect an incoming and outgoing call. This parameter determines if the outgoing call (from the device to the PBX) must be connected before the transfer is initiated. ▪ **[0]** Alert = Enable ISDN Transfer if outgoing call is in Alert or Connect state (default). ▪ **[1]** Connect = Enable ISDN Transfer only if outgoing call is in Connect state.
Remove CLI when Restricted **[RemoveCLIWhenRestricted]**	Determines (for IP-to-Tel calls) whether the Calling Number and Calling Name IEs are removed from the ISDN SETUP message if the presentation is set to Restricted. ▪ **[0]** No = IE aren't removed (default). ▪ **[1]** Yes = IE are removed.
Remove Calling Name **[RemoveCallingName]**	Enables the device to remove the Calling Name from SIP-to-ISDN calls. ▪ **[0]** Disable = Does not remove Calling Name (default). ▪ **[1]** Enable = Remove Calling Name.
Default Cause Mapping From ISDN to SIP **[DefaultCauseMapISDN2IP]**	Defines a single default ISDN release cause that is used (in ISDN-to-IP calls) instead of all received release causes, except when the following Q.931 cause values are received: Normal Call Clearing (16), User Busy (17), No User Responding (18), or No Answer from User (19). The range is valid Q.931 release causes (0 to 127). The default value is 0 (i.e., not configured - static mapping is used).
Add Prefix to Redirect Number **[Prefix2RedirectNumber]**	Defines a string prefix that is added to the Redirect number received from the Tel side. This prefix is added to the Redirect Number in the Diversion header. The valid range is an 8-character string. The default is an empty string.

Parameter	Description
Enable Calling Party Category **[EnableCallingPartyCategory]**	Determines whether Calling Party Category (CPC) is mapped between SIP and PRI. ▪ **[0]** Disable = Don't relay the CPC between SIP and PRI (default). ▪ **[1]** Enable = The CPC is relayed between SIP and PRI. If enabled, the CPC received in the Originating Line Information (OLI) IE of an incoming ISDN SETUP message is relayed to the From / P-Asserted-Identity headers using the 'cpc' parameter, in the outgoing INVITE message, and vice versa. For example (calling party is a payphone): From:<sip:2000;cpc=payphone@10.8.23.70>;tag=1c1806157451 **Note:** This feature is supported only when using the NI-2 PRI variant.
Digital Out-Of-Service Behavior **[DigitalOOSBehavior]**	Determines the method for setting digital trunks to Out-Of-Service state per device. ▪ **[0]** Default = Uses default behavior for each trunk - see note below (default) ▪ **[1]** Service = Sends ISDN In or Out of Service (only for ISDN protocols that support Service message). ▪ **[2]** D-Channel = Takes D-Channel down or up (ISDN only). ▪ **[3]** Alarm = Sends or cleans PSTN AIS Alarm (ISDN and CAS). ▪ **[4]** Block = Blocks trunk (CAS only). **Notes:** ▪ The default behavior (value 0) is as follows: - ISDN: Use Service messages on supporting variants and use Alarm on non-supporting variants. - CAS: Use Alarm. ▪ When updating this parameter value at run-time, you must stop the trunk and then restart it for the update to take effect. ▪ To determine the method for setting Out-Of-Service state per trunk, use the DigitalOOSBehaviorFor Trunk_ID parameter (refer to "Trunk Settings" on page 96).
MLPP (Multilevel Precedence and Preemption)	
Call Priority Mode **[CallPriorityMode]**	Enables Priority Calls handling. ▪ **[0]** Disable = Disable (default). ▪ **[1]** MLPP = Priority Calls handling is enabled.
MLPP Default Namespace **[MLPPDefaultNamespace]**	Determines the Namespace used for MLPP calls received from the ISDN side and destined for the Application server. The Namespace value is not present in the Precedence IE of the PRI SETUP message. Therefore, the value is used in the Resource-Priority header of the outgoing SIP INVITE request. ▪ **[1]** DSN = DSN (default) ▪ **[2]** DOD = DOD ▪ **[3]** DRSN = DRSN

Parameter	Description
Default Call Priority **[SIPDefaultCallPriority]**	Defines the default call priority for MLPP calls. ▪ **[0]** 0 = ROUTINE (default) ▪ **[2]** 2 = PRIORITY ▪ **[6]** 6 = IMMEDIATE ▪ **[8]** 8 = FLASH-OVERRIDE ▪ **[9]** 9 = FLASH-OVERRIDE-OVERRIDE If the incoming SIP INVITE request doesn't contain a valid priority value in the SIP Resource-Priority header, the default value is used in the Precedence IE (after translation to the relevant ISDN Precedence value) of the outgoing PRI SETUP message. If the incoming PRI SETUP message doesn't contain a valid Precedence Level value, the default value is used in the Resource-Priority header of the outgoing SIP INVITE request. In this scenario, the character string is sent without translation to a numerical value.
MLPP DiffServ **[MLPPDiffserv]**	Defines the DiffServ value (differentiated services code point -- DSCP) used in IP packets containing SIP messages that are related to MLPP calls. The valid range is 0 to 63. The default value is 50.

3.4.5.9 Configuring the IPmedia Parameters

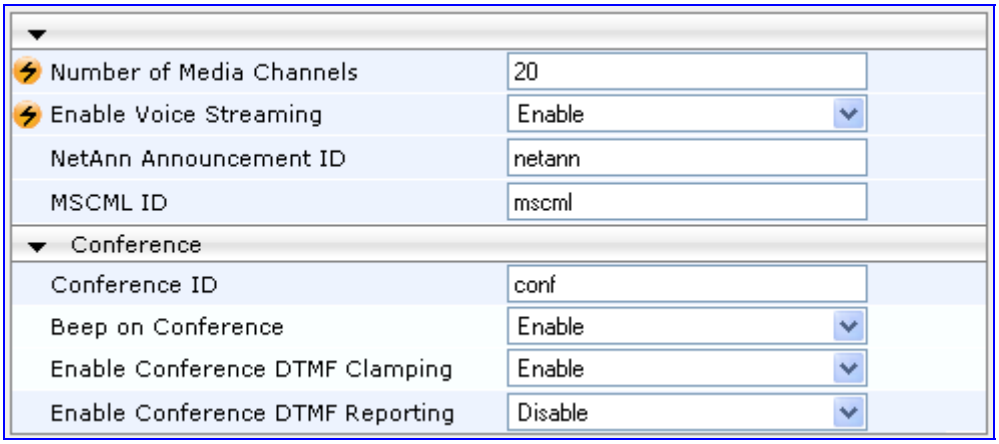
The 'IPmedia Parameters' page allows you to configure the IP media parameters.

Note: The 'IPmedia Parameters' page is applicable only to Mediant 1000. The 'IPmedia Parameters' page is Feature Key dependant and therefore is available only if included in the device's Feature Key (refer to 'Updating the Software Upgrade Key' on page 268).

➢ To configure the IP media parameters, take these 4 steps:

1. Open the 'IPmedia Parameters' page (**Configuration** tab > **Protocol Configuration** menu > **IP Media** submenu > **IPMedia Parameters** page item).

Figure 3-97: IPmedia Parameters Page



2. Configure the IP media parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265**.**

Table 3-55: IP Media Configuration Parameters

***ini* File Field Name** **Web Parameter Name**	**Valid Range and Description**
Number of Media Channels **[MediaChannels]**	The number of DSP channels that are allocated for IP conferencing, IP streaming, and IP transcoding (other DSP channels can be used for PSTN device). The maximum value of Media Channels depends on the number of installed Media Processing modules (MPM): 1 module = 20 channels; 2 modules = 60; 3 modules = 100. The default value is 0.
Enable Voice Streaming **[EnableVoiceStreaming]**	Enables/disables the HTTP Voice Streaming application (play / record). ▪ **[0]** Disable = Voice Streaming is disabled (default). ▪ **[1]** Enable = Voice Streaming is enabled.

ini File Field Name Web Parameter Name	Valid Range and Description
NetAnn Announcement ID **[NetAnnAnncID]**	NetAnn identification string (up to 16 characters) for playing an announcement using the NetAnn interface. The application server sends a regular SIP INVITE message with a SIP URI that includes this identifier string and a “play=” parameter that identifies the necessary announcement. The default value is ‘annc’. Example 1: INVITE sip: annc@10.2.3.4;play=http://localhost/1. Example 2: INVITE sip: annc@10.2.3.4;play=http://10.2.3.4/Annc/hello.wav.
MSCML ID **[MSCMLID]**	Media Server Control Markup Language (MSCML) identification string (up to 16 characters). To start an MSCML session, the application server sends a regular SIP INVITE message with a SIP URI that includes this string. The default value is ‘ivr’. For example: INVITE sip:ivr@10.2.3.4 Subsequent INFO messages carry the requests and responses.
Transcoding ID **[TranscodingID]**	Transcoding identification string (up to 16 characters) used for identifying an incoming Transcoding call. The default value is ‘trans’. For detailed information on Transcoding, refer to "NetAnn Interface" on page 483.
Conference	
Conference ID **[ConferenceID]**	Conference Identification string (up to 16 characters). The default value is ‘conf’. For example: ConferenceID = MyConference **Note:** To join a conference, the INVITE URI must include the Conference ID string, preceded by the number of the participants in the conference, and terminated by a unique number. For example: Invite sip:4MyConference1234@10.1.10.10. INVITE messages with the same URI join the same conference.
Beep on Conference **[BipOnConference]**	Determines whether or not a beep is played when a participant joins or leaves a conference (in the latter case, a beep of a different pitch is heard). ▪ **[0]** Disable = Beep is disabled. ▪ **[1]** Enable = Beep is enabled (default).
Enable Conference DTMF Clamping **[EnableConferenceDTMFClamp]**	Determines the device logic once a DTMF is received on any conference participant. If enabled, the DTMF is not regenerated toward the other conference participants. This logic is only relevant for simple conferencing (NetAnn). ▪ **[0]** Disable = Disable ▪ **[1]** Enable = Enable (default)
Enable Conference DTMF Reporting **[EnableConferenceDTMFReporting]**	Determines the device logic once a DTMF is received on any conference participant. If enabled, the device reports this DTMF in an out-of-band SIP message (according to TxDTMFOptions). This logic is only relevant for simple conferencing (NetAnn). ▪ **[0]** Disable = Disable (default) ▪ **[1]** Enable = Enable

3.4.6 Advanced Applications

The **Advanced Applications** menu allows you to configure advanced SIP-based applications. This menu includes the following page items:

- Voice Mail Settings (refer to "Configuring the Voice Mail Parameters" on page 244)
- RADIUS Parameters (refer to "Configuring RADIUS Accounting Parameters" on page 247)
- FXO Settings (refer to Configuring the FXO Parameters on page 248)

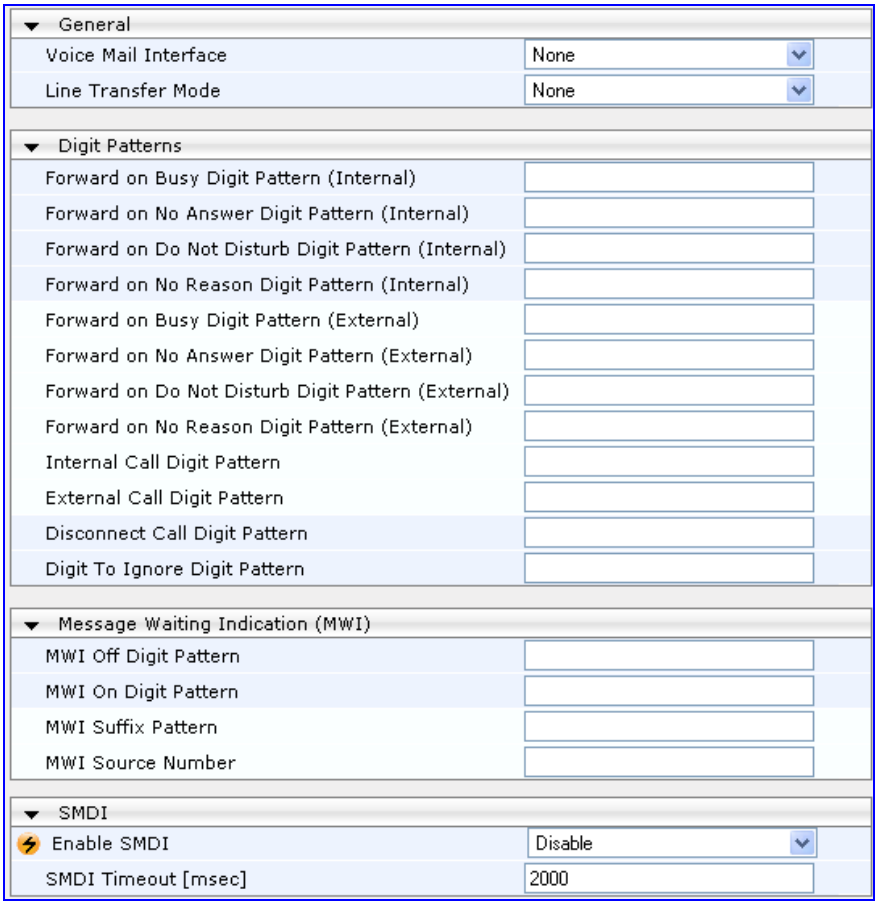
3.4.6.1 Configuring the Voice Mail (VM) Parameters

The 'Voice Mail Settings' page allows you to configure the voice mail parameters. The voice mail application applies only to FXO and CAS interfaces. For detailed information on voice mail, refer to the *CPE Configuration Guide for Voice Mail User's Manual*.

➢ To configure the Voice Mail parameters, take these 4 steps:

1. Open the 'Voice Mail Settings' page (**Configuration** tab > **Advanced Applications** menu > **Voice Mail Settings** page item).

Figure 3-98: Voice Mail Settings Page



2. Configure the voice mail parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-56: Voice Mail Parameters

Parameter	Description
General	
Voice Mail Interface **[VoiceMailInterface]**	Enables the voice mail application on the device and determines the communication method used between the PBX and the device. ▪ **[0]** None (default) ▪ **[1]** DTMF ▪ **[2]** SMDI (N/A)[3] QSIG ▪ [4] SETUP Only (ISDN) ▪ [5] MATRA/AASTRA QSIG
Line Transfer Mode **[LineTransferMode]**	Determines the call transfer method used by the device. ▪ **[0]** None = IP (default). ▪ **[1]** Blind = PBX blind transfer. After receiving a REFER message from the IP side, the FXO sends a hook-flash to the PBX, dials the digits (that are received in the Refer-To header), and then immediately drops the line (on-hook). The PBX performs the transfer internally. ▪ **[2]** Semi Supervised = PBX Semi-Supervised transfer. After receiving a REFER message from the IP side, the FXO sends a hook-flash to the PBX, and then dials the digits (that are received in the Refer-To header). If no Busy or Reorder tones are detected (within approximately 2 seconds), the device completes the call transfer by releasing the line; otherwise, the transfer is cancelled, the device sends a SIP NOTIFY message with a failure reason in the NOTIFY body (such as 486 if busy tone detected), and generates an additional hook-flash towards the FXO line to restore connection to the original call. ▪ **[3]** Supervised = PBX Supervised transfer. After receiving a REFER message from the IP side, the FXO sends a hook-flash to the PBX, and then dials the digits (that are received in the Refer-To header). The FXO waits for connection of the transfer call and if speech is detected (e.g., "hello") within approximately 2 seconds, the device completes the call transfer by releasing the line; otherwise, the transfer is cancelled, the device sends a SIP NOTIFY message with a failure reason in the NOTIFY body (such as 486 if busy tone detected), and generates an additional hook-flash towards the FXO line to restore connection to the original call. **Note:** This parameter is applicable only to Mediant 1000.
Digit Patterns The following digit pattern parameters apply only to voice mail applications that use the DTMF communication method. For the available patterns' syntaxes, refer to the *CPE Configuration Guide for Voice Mail*.	
Forward on Busy Digit Pattern (Internal) **[DigitPatternForwardOnBusy]**	Determines the digit pattern used by the PBX to indicate 'call forward on busy' when the original call is received from an internal extension. The valid range is a 120-character string.

Parameter	Description
Forward on No Answer Digit Pattern (Internal) **[DigitPatternForwardOnNoAnswer]**	Determines the digit pattern used by the PBX to indicate 'call forward on no answer' when the original call is received from an internal extension. The valid range is a 120-character string.
Forward on Do Not Disturb Digit Pattern (Internal) **[DigitPatternForwardOnDND]**	Determines the digit pattern used by the PBX to indicate 'call forward on do not disturb' when the original call is received from an internal extension. The valid range is a 120-character string.
Forward on No Reason Digit Pattern (Internal) **[DigitPatternForwardNoReason]**	Determines the digit pattern used by the PBX to indicate 'call forward with no reason' when the original call is received from an internal extension. The valid range is a 120-character string.
Forward on Busy Digit Pattern (External) **[DigitPatternForwardOnBusyExt]**	Determines the digit pattern used by the PBX to indicate 'call forward on busy' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Forward on No Answer Digit Pattern (External) **[DigitPatternForwardOnNoAnswerExt]**	Determines the digit pattern used by the PBX to indicate 'call forward on no answer' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Forward on Do Not Disturb Digit Pattern (External) **[DigitPatternForwardOnDNDExt]**	Determines the digit pattern used by the PBX to indicate 'call forward on do not disturb' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Forward on No Reason Digit Pattern (External) **[DigitPatternForwardNoReasonExt]**	Determines the digit pattern used by the PBX to indicate 'call forward with no reason' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Internal Call Digit Pattern **[DigitPatternInternalCall]**	Determines the digit pattern used by the PBX to indicate an internal call. The valid range is a 120-character string.
External Call Digit Pattern **[DigitPatternExternalCall]**	Determines the digit pattern used by the PBX to indicate an external call. The valid range is a 120-character string.
Disconnect Call Digit Pattern **[TelDisconnectCode]**	Determines a digit pattern that when received from the Tel side, indicates the device to disconnect the call. The valid range is a 25-character string.
Digit To Ignore Digit Pattern **[DigitPatternDigitToIgnore]**	A digit pattern that if received as Src (S) or Redirect (R) numbers is ignored and not added to that number. The valid range is a 25-character string.
Message Waiting Indication (MWI)	
MWI Off Digit Pattern **[MWIOffCode]**	Determines the digit code used by the device to notify the PBX that there aren't any messages waiting for a specific extension. This code is added as prefix to the dialed number. The valid range is a 25-character string.
MWI On Digit Pattern **[MWIOnCode]**	Determines the digit code used by the device to notify the PBX of messages waiting for a specific extension. This code is added as prefix to the dialed number. The valid range is a 25-character string.

Parameter	Description
MWI Suffix Pattern **[MWISuffixCode]**	Determines the digit code used by the device as a suffix for 'MWI On Digit Pattern' and 'MWI Off Digit Pattern'. This suffix is added to the generated DTMF string after the extension number. The valid range is a 25-character string.
SMDI (currently not supported)	
Enable SMDI **[SMDI]**	N/A
SMDI Timeout **[SMDITimeOut]**	N/A

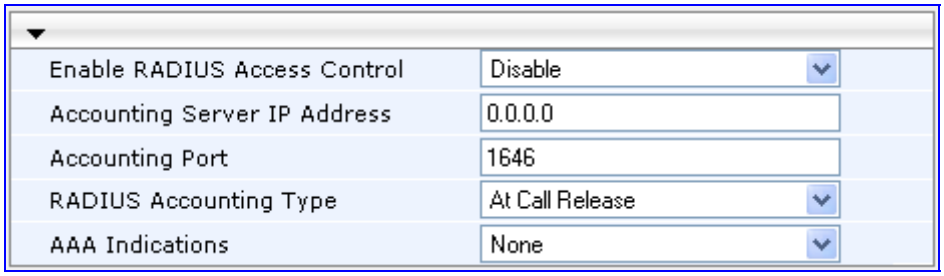
3.4.6.2 Configuring RADIUS Accounting Parameters

The 'RADIUS Parameters' page is used for configuring the Remote Authentication Dial In User Service (RADIUS) accounting parameters.

➢ To configure the RADIUS parameters, take these 4 steps:

1. Open the 'RADIUS Parameters' page (**Configuration** tab > **Advanced Applications** menu > **RADIUS Parameters** page item).

Figure 3-99: RADIUS Parameters Page



2. Configure the RADIUS accounting parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-57: RADIUS Parameters Description

Parameter	Description
Enable RADIUS Access Control **EnableRADIUS**	Enables or disables the RADIUS application. ▪ **[0]** Disables = disables RADIUS application (default) ▪ **[1]** Enable = enables RADIUS application
Accounting Server IP Address **[RADIUSAccServerIP]**	IP address of the RADIUS accounting server.
Accounting Port **[RADIUSAccPort]**	Port of the RADIUS accounting server. The default value is 1646.

Parameter	Description
RADIUS Accounting Type **[RADIUSAccountingType]**	Determines when the RADIUS accounting messages are sent to the RADIUS accounting server. ▪ **[0]** At Call Release = Sent at call release only (default). ▪ **[1]** At Connect & Release = Sent at call connect and release. ▪ **[2]** At Setup & Release = Sent at call setup and release.
AAA Indications **[AAAIndications]**	Determines the Authentication, Authorization and Accounting (AAA) indications. ▪ **[0]** None = No indications (default). ▪ **[3]** Accounting Only = Only accounting indications are used.

3.4.6.3 Configuring the FXO Parameters

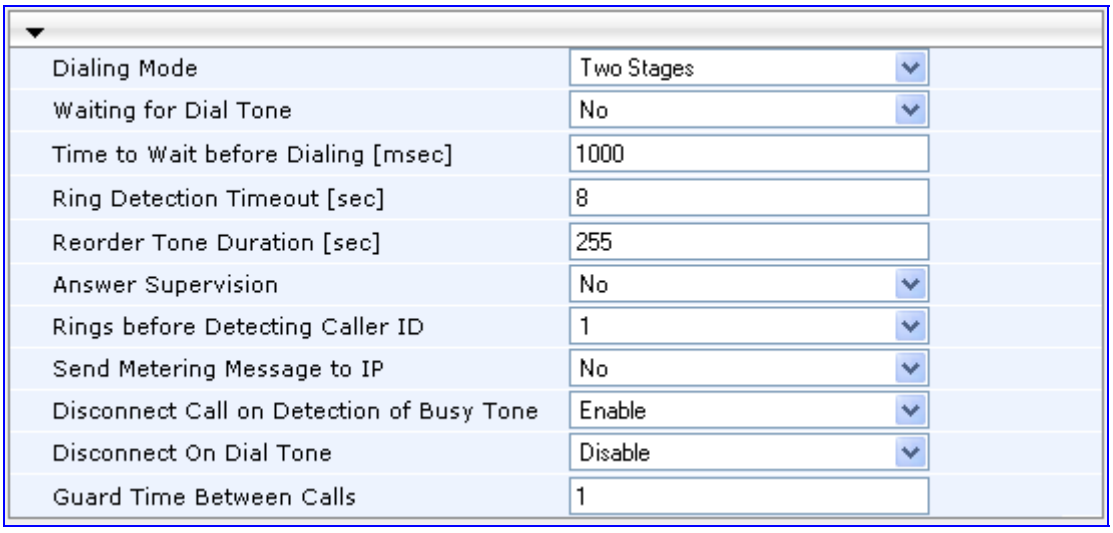
The 'FXO Settings' page allows you to configure the device's specific FXO parameters.

Note: The 'FXO Settings' page is only available for the FXO interfaces and applicable to Mediant 1000.

➢ **To configure the FXO parameters, take these 4 steps:**

1. Open the 'FXO Settings' page (**Configuration** tab > **Advanced Applications** menu > **FXO Settings** page item).

Figure 3-100: FXO Settings Page



2. Configure the FXO parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-58: FXO Parameters Description

Parameter	Description
Dialing Mode **[IsTwoStageDial]**	Determines the dialing mode for IP-to-Tel (FXO) calls. ▪ **[0]** One Stage = One-stage dialing. ▪ **[1]** Two Stages = Two-stage dialing (default). If two-stage dialing is enabled, the device seizes one of the PSTN/PBX lines without performing any dialing, connects the remote IP user to the PSTN/PBX, and all further signaling (dialing and Call Progress Tones) is performed directly with the PBX without the device's intervention. If one-stage dialing is enabled, the device seizes one of the available lines (according to the parameter ChannelSelectMode), and dials the destination phone number received in the INVITE message. Use the parameter IsWaitForDialTone to specify whether the dialing must start after detection of the dial tone or immediately after seizing the line.
Waiting For Dial Tone **[IsWaitForDialTone]**	Determines whether the device waits for a dial tone before dialing the phone number for IP-to-Tel (FXO) calls. ▪ **[0]** No = Don't wait for dial tone. ▪ **[1]** Yes = Wait for dial tone (default). When one-stage dialing and this parameter are enabled, the device dials the phone number (to the PSTN/PBX line) only after it detects a dial tone. If this parameter is disabled, the device immediately dials the phone number after seizing the PSTN/PBX line without 'listening' for a dial tone. **Notes:** ▪ The correct dial tone parameters should be configured in the Call Progress Tones file. ▪ The device may take 1 to 3 seconds to detect a dial tone (according to the dial tone configuration in the Call Progress Tones file).
Time to Wait before Dialing [msec] **[WaitForDialTime]**	For Digital: Determines the delay after hook-flash is generated and until dialing begins. Applies to call transfer (i.e., the parameter TrunkTransferMode is set to 3) on CAS devices. For Analog: Determines the delay before the device starts dialing on the FXO line in the following scenarios: ▪ The delay between the time the line is seized and dialing begins during the establishment of an IP-to-Tel call. **Note:** Applicable only for one-stage dialing, when the parameter IsWaitForDialTone is disabled. ▪ The delay between detection of a Wink and the start of dialing during the establishment of an IP-to-Tel call (for DID lines, EnableDIDWink is set to 1). ▪ For call transfer - the delay after hook-flash is generated and dialing begins. The valid range (in milliseconds) is 0 to 20,000 (i.e., 20 seconds). The default value is 1,000 (i.e., 1 second).

Parameter	Description
Ring Detection Timeout [sec] **[FXOBetweenRingTime]**	Defines the timeout (in seconds) for detecting the second ring after the first detected ring. If automatic dialing is not used and Caller ID is enabled, the device seizes the line after detection of the second ring signal (allowing detection of caller ID sent between the first and the second rings). If the second ring signal is not received within this timeout, the device doesn't initiate a call to IP. If automatic dialing is used, the deviceinitiates a call to IP when the ringing signal is detected. The FXO line is seized only if the remote IP party answers the call. If the remote party doesn't answer the call and the second ring signal is not received within this timeout, the device releases the IP call. This parameter is typically set to between 5 and 8. The default is 8. **Note:** This parameter is applicable only for Tel-to-IP calls.
Reorder Tone Duration [sec] **[TimeForReorderTone]**	For Analog: Busy or Reorder tone duration (in seconds) that the device plays before releasing the line. The valid range is 0 to 254. The default is 0 seconds. Typically, after playing a Reorder / Busy tone for the specified duration, the device starts playing an Offhook Warning tone. **Notes:** ▪ Selection of Busy or Reorder tone is performed according to the release cause received from IP. ▪ Refer also to the parameter CutThrough, described in "Advanced Parameters" on page 166. For Digital: Busy or Reorder Tone duration (in seconds) that the CAS device plays before releasing the line. The valid range is 0 to 254. The default value is 10. This parameter is also applicable for ISDN if PlayBusyTone2ISDN = 2. The selection of Busy or Reorder tone is done according to the release cause received from IP.
Answer Supervision **[EnableVoiceDetection]**	Enables sending of 200 OK upon detection of speech, fax, or modem. ▪ **[1]** Yes = device sends 200 OK (to INVITE) messages when speech/fax/modem is detected. ▪ **[0]** No = 200 OK is sent only once the device completes dialing (default). Typically, this feature is used only when early media (EnableEarlyMedia) is used to establish the voice path before the call is answered. **Note:** This feature is applicable only for one-stage dialing.
Rings before Detecting Caller ID **[RingsBeforeCallerID]**	Determines the number of rings before the device starts detecting Caller ID. ▪ **[0]** 0 = Before first ring. ▪ **[1]** 1 = After first ring (default). ▪ **[2]** 2 = After second ring.
Send Metering Message to IP **[SendMetering2IP]**	N/A.

Parameter	Description
Disconnect Call on Detection of Busy Tone **[DisconnectOnBusyTone]**	▪ **[0]** Enable = Do not disconnect call on detection of busy tone. ▪ **[1]** Disable = Call is released if busy or reorder (fast busy) tones are detected on the device's FXO port (default).
Disconnect on Dial Tone **[DisconnectOnDialTone]**	The device can disconnect a call after a dial tone is detected from the PBX. ▪ **[0]** Disable = Call isn't released. ▪ **[1]** Enable = Call is released if dial tone is detected on the device's FXO port (default). **Note:** This option is in addition to the mechanism that disconnects a call when either busy or reorder tones are detected.
Guard Time Between Calls **[GuardTimeBetweenCalls]**	Defines the time interval (in seconds) after a call has ended and a new call can be accepted for IP-to-Tel (FXO) calls. The valid range is 0 to 10. The default value is 1. **Note:** Occasionally, after a call ends and on-hook is applied, a delay is required before placing a new call (and performing off-hook). This is necessary to prevent incorrect hook-flash detection or other glare phenomena.

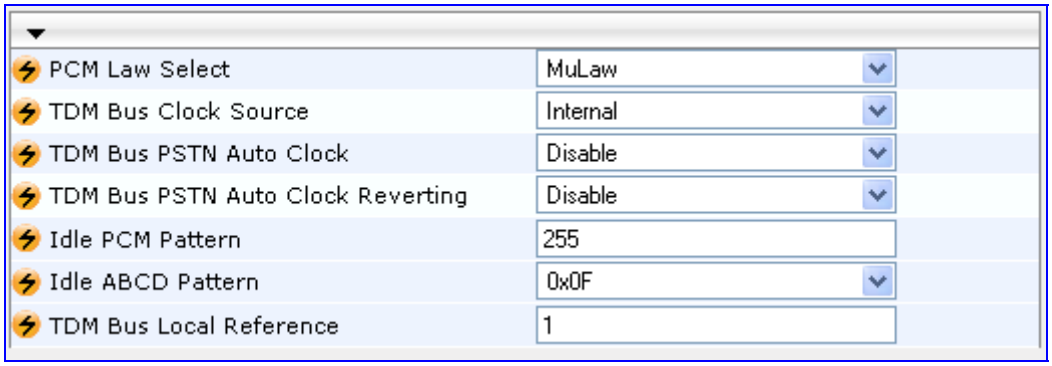
3.4.7 Configuring the TDM Bus Settings

The device's Time-Division Multiplexing (TDM) bus settings can be performed in the 'TDM Bus Settings' page, as described in the procedure below.

➢ To configure the TDM Bus settings, take these 5 steps:

1. Open the 'TDM Bus Settings' page (**Configuration** tab > **TDM Configuration** menu > **TDM Bus Settings** page item).

Figure 3-101: TDM Bus Settings Page



2. Configure the TDM bus parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. Save the changes to flash memory, refer to "Saving Configuration" on page 265.
5. Reset the device (refer to "Resetting the Device" on page 262).

Table 3-59: TDM Bus Settings Parameters Description

Parameter	Description
PCM Law Select **[PCMLawSelect]**	Determines the type of PCM companding law in input/output TDM bus. ▪ **[1]** Alaw = Alaw (default) ▪ **[3]** MuLaw = MuLaw **Note:** Typically, A-Law is used for E1 spans and μ-Law for T1/J1 spans.
Idle PCM Pattern **[IdlePCMPattern]**	Defines the PCM Pattern that is applied to the E1/T1 timeslot (B-channel) when the channel is idle. The range is 0 to 255. The default is set internally according to the Law select 1 (0xFF for Mu-Law; 0x55 for A-law).
Idle ABCD Pattern **[IdleABCDPattern]**	Defines the ABCD (CAS) Pattern that is applied to the CAS signaling bus when the channel is idle. The valid range is 0x0 to 0xF. The default is -1 (i.e., default pattern = 0000). **Note:** This parameter is applicable only when using PSTN interface with CAS protocols.
TDM Bus Local Reference **[TDMBusLocalReference]**	Physical Trunk ID from which the device recovers (receives) its clock synchronization. The range is 0 to maximum number of Trunks. The default is Trunk ID 1. **Note:** This parameter is applicable only if the parameter TDMBusClockSource is set to 4 and the parameter TDMBusPSTNAutoClockEnable is set to 0.
TDM Bus PSTN Auto Clock **[TDMBusPSTNAutoClockEnable]**	Enables or disables the PSTN trunk auto-fallback clock feature. ▪ **[0]** Disable (default) = Recovers the clock from the E1/T1 line defined by the parameter TDMBusLocalReference. ▪ **[1]** Enable = Recovers the clock from any connected synchronized slave E1/T1 line. If this trunk loses its synchronization, the device attempts to recover the clock from the next trunk. Note that initially, the device attempts to recover the clock from the trunk defined by the parameter TDMBusLocalReference. **Note:** This parameter is relevant only if the parameter TDMBusClockSource is set to 4.

Parameter	Description
TDM Bus PSTN Auto Clock Reverting **[TDMBusPSTNAutoClockRevertingEnable]**	Enables or disables the PSTN trunk auto-fallback reverting feature. If enabled and a trunk returning to service has an AutoClockTrunkPriority parameter value (refer to "Configuring the Trunk Settings" on page 96) that is higher than the priority of the local reference trunk (set in the TDMBusLocalReference parameter), the local reference reverts to the trunk with the higher priority that has returned to service for the device's clock source. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** This parameter is applicable only when the TDMBusPSTNAutoClockEnable parameter is set to 1.
TDM Bus Clock Source **[TDMBusClockSource]**	Selects the clock source to which the device synchronizes. ▪ **[1]** Internal = Generate clock from local source (default). ▪ **[4]** Network = Recover clock from PSTN line. For detailed information on configuring the device's clock settings, refer to "Clock Settings" on page 459.

3.5 Management Tab

The **Management** tab on the Navigation bar displays all menus related to device management. These menus appear in the Navigation tree and include the following:

- Management Configuration (refer to "Management Configuration" on page 253)
- Software Update (refer to "Software Update" on page 266)

3.5.1 Management Configuration

The **Management Configuration** menu allows you to configure the device's management parameters. This menu contains the following page items:

- Management Settings (refer to "Configuring the Management Settings" on page 254)
- Regional settings (refer to "Configuring the Regional Settings" on page 261)
- Maintenance Actions (refer to "Maintenance Actions" on page 262)

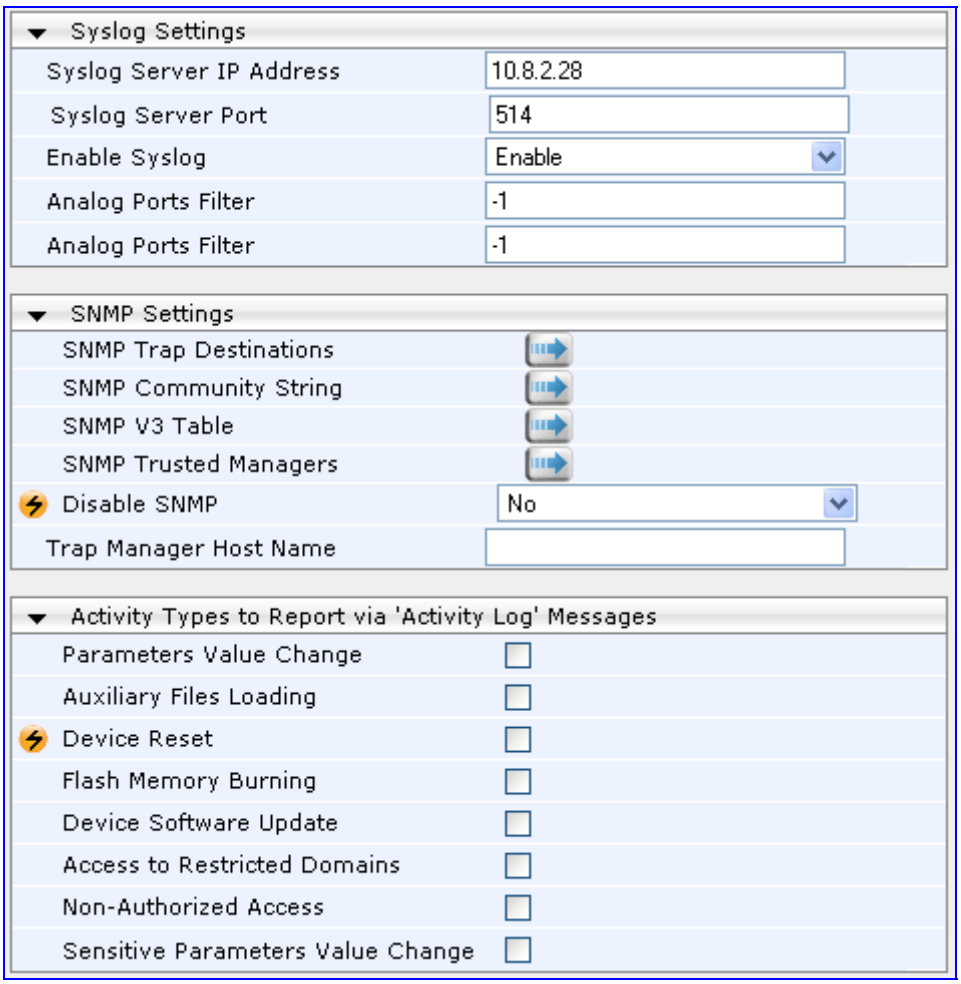
3.5.1.1 Configuring the Management Settings

The 'Management Settings' page allows you to configure the device's management parameters.

➢ To configure the Management parameters, take these 4 steps:

1. Open the 'Management Settings' page (**Management** tab > **Management Configuration** menu > **Management Settings** page item).

Figure 3-102: Management Settings Page



2. Configure the Management Settings according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Table 3-60: Management Settings Parameters

Parameter	Description
Syslog Settings	
Syslog Server IP Address **[SyslogServerIP]**	IP address (in dotted-decimal notation) of the computer you are using to run the Syslog server. The Syslog server is an application designed to collect the logs and error messages generated by the device. Default IP address is 0.0.0.0. For information on Syslog, refer to the *Product Reference Manual*.
Syslog Server Port **[SyslogServerPort]**	Defines the UDP port of the Syslog server. The valid range is 0 to 65,535. The default port is 514. For information on the Syslog, refer to the *Product Reference Manual*.
Enable Syslog **[EnableSyslog]**	Sends the logs and error message generated by the device to the Syslog server. ▪ **[0]** Disable = Logs and errors are not sent to the Syslog server (default). ▪ **[1]** Enable = Enables the Syslog server. **Notes:** ▪ If you enable Syslog, you must enter an IP address and a port number (using SyslogServerIP and SyslogServerPort parameters). ▪ You can configure the device to send Syslog messages implementing Debug Recording (refer to Debug Recording (DR)), by using the SyslogOutputMethod *ini* file parameter. ▪ Syslog messages may increase the network traffic. ▪ To configure Syslog logging levels, use the parameter GwDebugLevel, as described in "Advanced Parameters" on page 166. ▪ For information on the Syslog, refer to the *Product Reference Manual*. ▪ Logs are also sent to the RS-232 serial port. For information on establishing a serial communications link with the device, refer to the device's Fast Track Guide.
SNMP Settings For detailed information on the SNMP parameters that can be configured via the *ini* file, refer to "SNMP Parameters" on page 321. For detailed information on developing an SNMP-based program to manage your device, refer to the *Product Reference Manual*.	
SNMP Trap Destinations	Click the arrow button to configure the SNMP trap destinations (refer to "Configuring the SNMP Trap Destinations Table" on page 257).
SNMP Community String	Click the arrow button to configure the SNMP community strings (refer to "Configuring the SNMP Community Strings" on page 258).
SNMP V3 Table	Click the arrow button to configure the SNMP V3 users (refer to "Configuring SNMP V3 Table" on page 259).
SNMP Trusted Managers	Click the arrow button to configure the SNMP Trusted Managers (refer to "Configuring SNMP Trusted Managers" on page 260).
Enable SNMP **[DisableSNMP]**	▪ **[0]** Enable = SNMP is enabled (default). ▪ **[1]** Disable = SNMP is disabled and no traps are sent.

Parameter	Description
Trap Manager Host Name **[SNMPTrapManagerHostName]**	Defines an FQDN of a remote host that is used as an SNMP manager. The resolved IP address replaces the last entry in the Trap Manager table (defined by the parameter SNMPManagerTableIP_x) and the last trap manager entry of snmpTargetAddrTable in the snmpTargetMIB. For example: 'mngr.corp.mycompany.com'. The valid range is a 99-character string.
Activity Types to Report via 'Activity Log' Messages The Activity Log mechanism enables the device to send log messages (to a Syslog server) for reporting on certain types of Web operations according to the below user-defined filters.	
Parameters Value Change **[ActivityListToLog = PVC]**	Changes made on-the-fly to parameters.
Auxiliary Files Loading **[ActivityListToLog = AFL]**	Loading of auxiliary files (e.g., via 'Certificate' page).
Device Reset **[ActivityListToLog = DR]**	Reset of device via the 'Maintenance Actions' page.
Flash Memory Burning **[ActivityListToLog = FB]**	Burning of files / parameters to flash (e.g., 'Maintenance Actions' page).
Device Software Update **[ActivityListToLog = SWU]**	*cmp* loading via the Software Upgrade Wizard.
Access to Restricted Domains **[ActivityListToLog = ARD]**	Access to Restricted Domains, which includes the following pages: ▪ *ini* parameters (AdminPage) ▪ General Security Settings ▪ Configuration File ▪ IPSec/IKE tables ▪ Software Upgrade Key ▪ Internal Firewall ▪ Web Access List ▪ Web User Accounts
Non-Authorized Access **[ActivityListToLog = NAA]**	Attempt to access the Web interface with a false / empty user name or password.
Sensitive Parameters Value Change **[ActivityListToLog = SPC]**	Changes made to sensitive parameters: (1) IP Address (2) Subnet Mask (3) Default Gateway IP Address (4) ActivityListToLog

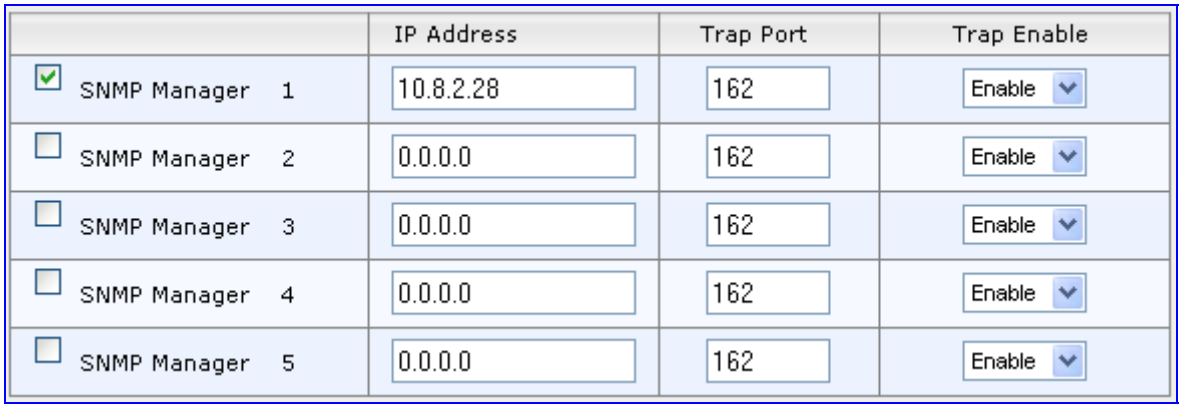
3.5.1.1.1 Configuring the SNMP Trap Destinations Table

The 'SNMP Trap Destinations' page allows you to configure up to five SNMP trap managers.

➢ To configure the SNMP Trap Destinations table, take these 5 steps:

1. Access the 'Management Settings' page, as described in "Configuring the Management Settings" on page 254.
2. In the 'SNMP Trap Destinations' field, click the right-pointing arrow button; the 'SNMP Trap Destinations' page appears.

Figure 3-103: SNMP Trap Destinations Page



3. Configure the SNMP trap managers parameters according to the table below.
4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Note: Only table row entries whose corresponding check boxes are selected are applied when clicking **Submit**; otherwise, settings revert to their defaults.

Table 3-61: SNMP Trap Destinations Parameters Description

Parameter	Description
SNMP Manager **[SNMPManagerIsUsed_x]**	Determines the validity of the parameters (IP address and port number) of the corresponding SNMP Manager used to receive SNMP traps. ▪ **[0]** (Check box cleared) = Disabled (default) ▪ **[1]** (Check box selected) = Enabled
IP Address **[SNMPManagerTableIP_x]**	IP address of the remote host used as an SNMP Manager. The device sends SNMP traps to these IP addresses. Enter the IP address in dotted-decimal notation, e.g., 108.10.1.255.
Trap Port **[SNMPManagerTrapPort_x]**	Defines the port number of the remote SNMP Manager. The device sends SNMP traps to these ports. The valid SNMP trap port range is 100 to 4000. The default port is 162.

Parameter	Description
Trap Enable **[SNMPManagerTrapSendingEnable_x]**	Activates or de-activates the sending of traps to the corresponding SNMP Manager. ▪ **[0]** Disable = Sending is disabled. ▪ **[1]** Enable = Sending is enabled (default).

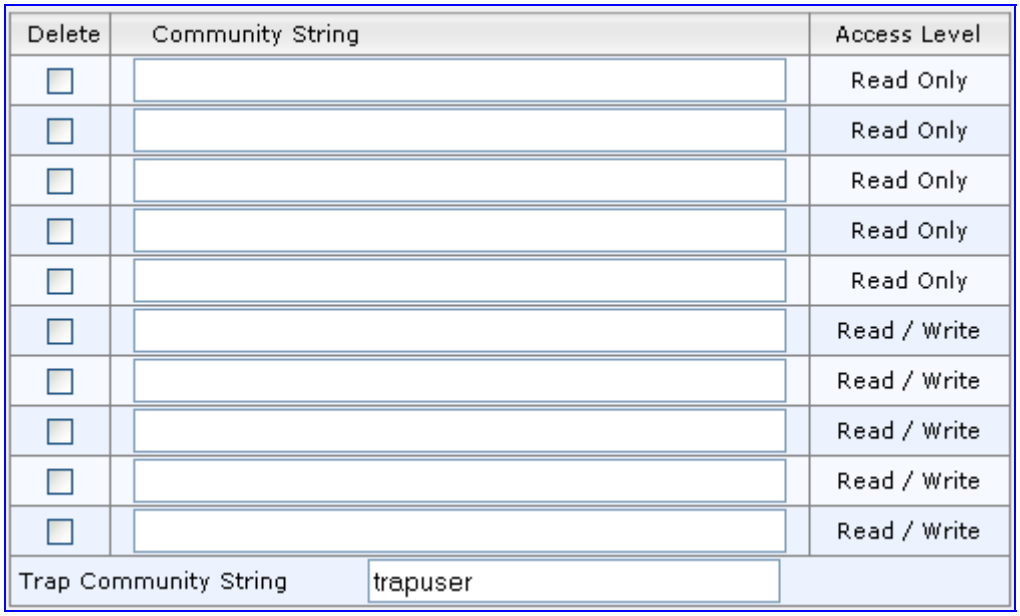
3.5.1.1.2 Configuring the SNMP Community Strings

The 'SNMP Community String' page allows you to configure up to five read-only and up to five read-write SNMP community strings, and to configure the community string that is used for sending traps. For detailed information on SNMP community strings, refer to the *Product Reference Manual*.

➢ To configure the SNMP community strings, take these 5 steps:

1. Access the 'Management Settings' page, as described in "Configuring the Management Settings" on page 254.
2. In the 'SNMP Community String' field, click the right-pointing arrow button; the 'SNMP Community String' page appears.

Figure 3-104: SNMP Community Strings Page



3. Configure the SNMP community strings parameters according to the table below.
4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 265.

Note: To delete a community string, select the **Delete** check box corresponding to the community string that you want to delete, and then click **Submit**.

Table 3-62: SNMP Community Strings Parameters Description

Parameter	Description
Community String	▪ Read Only **[SNMPReadOnlyCommunityString_x]:** Up to five read-only community strings (up to 19 characters each). The default string is 'public'. ▪ Read / Write **[SNMPReadWriteCommunityString_x]:** Up to five read / write community strings (up to 19 characters each). The default string is 'private'.
Trap Community String **[SNMPTrapCommunityString]**	Community string used in traps (up to 19 characters). The default string is 'trapuser'.

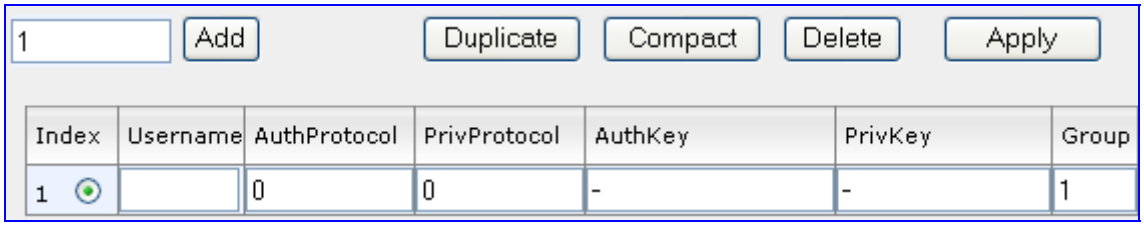
3.5.1.1.3 Configuring SNMP V3 Users

The 'SNMP V3 Settings' page allows you to configure authentication and privacy for up to 10 SNMP v3 users.

➢ To configure the SNMP v3 users, take the following 6 steps:

1. Access the 'Management Settings' page, as described in "Configuring the Management Settings" on page 254.
2. In the 'SNMP V3 Table' field, click the right-pointing arrow button; the 'SNMP V3 Settings' page appears.

Figure 3-105: SNMP V3 Setting Page



3. To add an SNMP v3 user, in the 'Add' field, enter the desired row index, and then click **Add**. A new row appears.
4. Configure the SNMP V3 Setting parameters according to the table below.
5. Click the **Apply** button to save your changes.
6. To save the changes, refer to "Saving Configuration" on page 265.

Notes:

- For a description of the web interface's table command buttons (e.g., **Duplicate** and **Delete**), refer to "Working with Tables" on page 36.
- You can also configure SNMP v3 users using the *ini* file table parameter SNMPUsers (refer to "SNMP Parameters" on page 321).

Table 3-63: SNMP V3 Users Parameters

Parameter	Description
Index **[SNMPUsers_Index]**	The table index. The valid range is 0 to 9.
Username **[SNMPUsers_Username]**	Name of the SNMP v3 user. This name must be unique.
AuthProtocol **[SNMPUsers_AuthProtocol]**	Authentication protocol of the SNMP v3 user. ▪ 0 = None (default) ▪ 1 = MD5 ▪ 2 = SHA-1
PrivProtocol **[SNMPUsers_PrivProtocol]**	Privacy protocol of the SNMP v3 user. ▪ 0 = none (default) ▪ 1 = DES ▪ 2 = 3DES ▪ 3 = AES128 ▪ 4 = AES192 ▪ 5 = AES256
AuthKey **[SNMPUsers_AuthKey]**	Authentication key. Keys can be entered in the form of a text password or long hex string. Keys are always persisted as long hex strings and keys are localized.
PrivKey **[SNMPUsers_PrivKey]**	Privacy key. Keys can be entered in the form of a text password or long hex string. Keys are always persisted as long hex strings and keys are localized.
Group **[SNMPUsers_Group]**	The group with which the SNMP v3 user is associated. ▪ 0 = read-only group (default) ▪ 1 = read-write group ▪ 2 = trap group **Note:** All groups can be used to send traps.

3.5.1.1.4 Configuring SNMP Trusted Managers

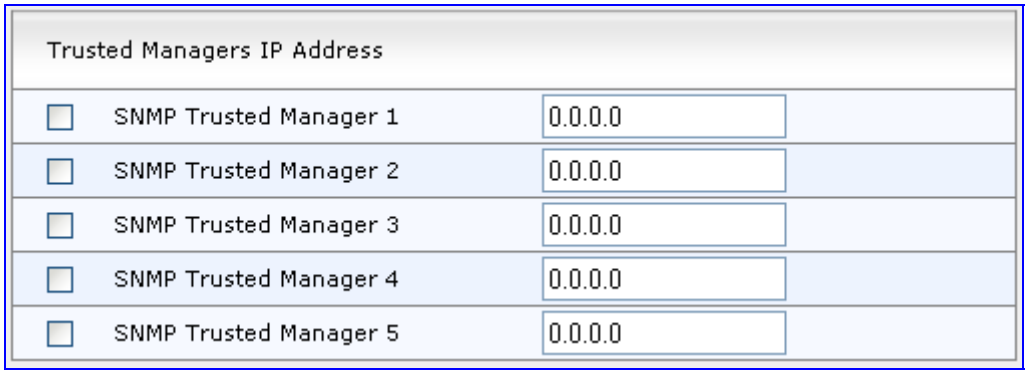
The 'SNMP Trusted Managers' page allows you to configure up to five SNMP Trusted Managers, based on IP addresses. By default, the SNMP agent accepts SNMP Get and Set requests from any IP address, as long as the correct community string is used in the request. Security can be enhanced by using Trusted Managers, which is an IP address from which the SNMP agent accepts and processes SNMP requests.

➢ **To configure the SNMP Trusted Managers, take the following 6 steps:**

1. Access the 'Management Settings' page, as described in "Configuring the Management Settings" on page 254.

2. In the 'SNMP Trusted Managers' field, click the right-pointing arrow button; the 'SNMP Trusted Managers' page appears.

Figure 3-106: SNMP Trusted Managers



3. Select the check box corresponding to the SNMP Trusted Manager that you want to enable and for whom you want to define an IP address.
4. Define an IP address in dotted-decimal notation.
5. Click the **Submit** button to apply your changes.
6. To save the changes, refer to "Saving Configuration" on page 265.

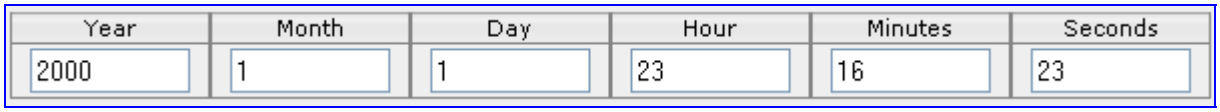
3.5.1.2 Configuring the Regional Settings

The 'Regional Settings' page allows you to define and view the device's internal date and time.

➢ To configure the device's date and time, take these 3 steps:

1. Open the 'Regional Settings' page (**Management** tab > **Management Configuration** menu > **Regional Settings** page item).

Figure 3-107: Regional Settings Page



2. Enter the current date and time in the geographical location in which the device is installed.
3. Click the **Submit** button; the date and time are automatically updated.

Notes:

- If the device is configured to obtain the date and time from an SNTP server (refer to "Configuring the Application Settings" on page 71), the fields on this page are read-only and cannot be modified. For an explanation on SNTP, refer to "Simple Network Time Protocol Support" on page 449.
- After performing a hardware reset, the date and time are returned to their defaults and therefore, should be updated.

3.5.1.3 Maintenance Actions

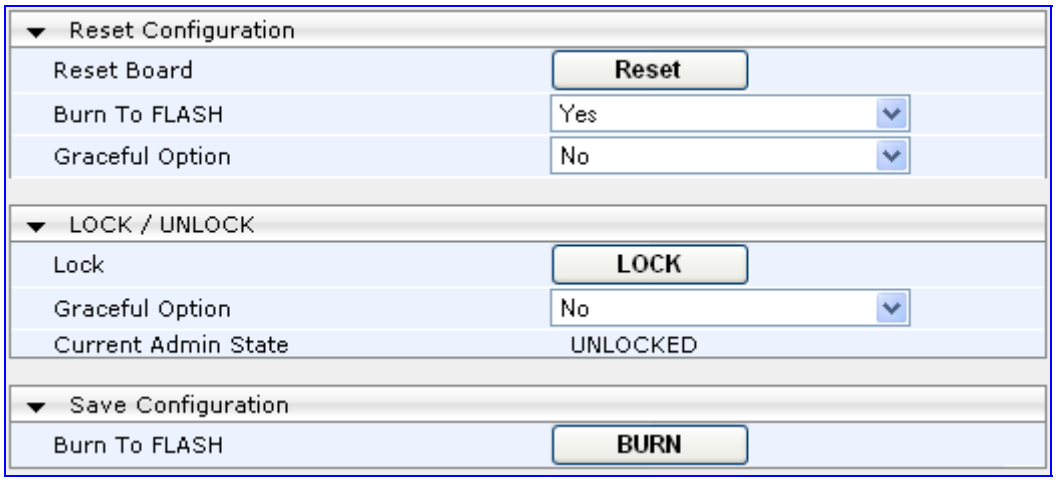
The 'Maintenance Actions' page allows you to perform the following operations:

- Reset the device (refer to "Resetting the Device" on page 262)
- Lock and unlock the device (refer to "Locking and Unlocking the Device" on page 264)
- Save the configuration to the device's flash memory (refer to "Saving Configuration" on page 265)

➢ To access the 'Maintenance Actions' page, take this step:

- On the Navigation bar, click the **Management** tab, and then in the Navigation tree, select the **Management Configuration** menu, and then choose the **Maintenance Actions** page item.

Figure 3-108: Maintenance Actions Page



3.5.1.3.1 Resetting the Device

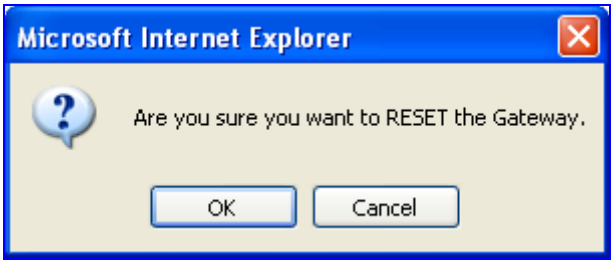
The 'Maintenance Actions' page allows you to remotely reset the device. In addition, before resetting the device, you can choose the following options:

- Save the device's current configuration to the device's flash memory (non-volatile).
- Perform a graceful shutdown, i.e., device reset starts only after a user-defined time expires (i.e., timeout) or after no more active traffic exists (the earliest thereof).

➢ To reset the device, take these 6 steps:

1. Open the 'Maintenance Actions' page (refer to "Maintenance Actions" on page 262).
2. Under the 'Reset Configuration' group, from the 'Burn To FLASH' drop-down list, select one of the following options:
   - 'Yes': The device's current configuration is saved (*burned*) to the flash memory prior to reset (default).
   - 'No': Resets the device without saving the current configuration to flash (discards all unsaved modifications).
3. Under the 'Reset Configuration' group, from the 'Graceful Option' drop-down list, select one of the following options:
   - 'Yes': Reset starts only after the user-defined time in the 'Shutdown Timeout' field (refer to Step 4) expires or after no more active traffic exists (the earliest thereof). In addition, no new traffic is accepted.
   - 'No': Reset starts regardless of traffic, and any existing traffic is terminated at once.
4. In the 'Shutdown Timeout' field (relevant only if the 'Graceful Option' in the previous step is set to 'Yes'), enter the time after which the device resets. Note that if no traffic exists and the time has not yet expired, the device resets.
5. Click the **Reset** button; a confirmation message box appears, requesting you to confirm.

Figure 3-109: Reset Confirmation Message Box



6. Click **OK** to confirm device reset; if the parameter 'Graceful Option' is set to 'Yes' (in Step 3), the reset is delayed and a screen displaying the number of remaining calls and time is displayed. When the device begins to reset, a message appears notifying you of this.

Notes:

- Throughout the Web interface, parameters preceded by the lightning symbol are not applied on-the-fly to the device and require that you reset the device for them to take effect.
- If you modify parameters that only take effect after a device reset, after you click the **Submit** button, the toolbar displays the word 'Reset' (refer to "Toolbar" on page 28) to remind you to later reset the device.

3.5.1.3.2 Locking and Unlocking the Device

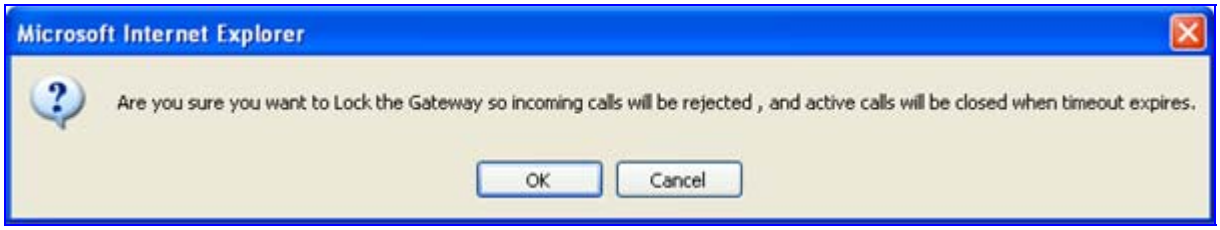
The Lock and Unlock options allow you to lock the device so that it doesn't accept any new incoming calls. This is useful when, for example, you are uploading new software files to the device and you don't want any traffic to interfere with the process.

➢ To lock the device, take these 5 steps:

1. Open the 'Maintenance Actions' page (refer to "Maintenance Actions" on page 262).
2. Under the 'LOCK / UNLOCK' group, from the 'Graceful Option' drop-down list, select one of the following options:
   - 'Yes': The device is 'locked' only after the user-defined time in the 'Lock Timeout' field (refer to Step 3) expires or no more active traffic exists (the earliest thereof). In addition, no new traffic is accepted.
   - 'No': The device is 'locked' regardless of traffic. Any existing traffic is terminated immediately.

   Note: These options are only available if the current status of the device is in the Unlock state.
3. In the 'Lock Timeout' field (relevant only if the parameter 'Graceful Option' in the previous step is set to 'Yes'), enter the time (in seconds) after which the device locks. Note that if no traffic exists and the time has not yet expired, the device locks.
4. Click the **LOCK** button; a confirmation message box appears requesting you to confirm device Lock.

Figure 3-110: Device Lock Confirmation Message Box


Microsoft Internet Explorer
Are you sure you want to Lock the Gateway so incoming calls will be rejected , and active calls will be closed when timeout expires.
OK
Cancel

5. Click **OK** to confirm device Lock; if 'Graceful Option' is set to 'Yes', the lock is delayed and a screen displaying the number of remaining calls and time is displayed. Otherwise, the lock process begins immediately. The 'Current Admin State' field displays the current state: LOCKED or UNLOCKED.

➢ To unlock the device, take these 2 steps:

1. Open the 'Maintenance Actions' page (refer to "Maintenance Actions" on page 262).
2. Under the 'LOCK / UNLOCK' group, click the **UNLOCK** button. Unlock starts immediately and the device accepts new incoming calls.

3.5.1.3.3 Saving Configuration

The 'Maintenance Actions' page allows you to save (*burn*) the current parameter configuration (including loaded auxiliary files) to the device's *non-volatile* memory (i.e., flash). The parameter modifications that you make throughout the Web interface's pages are temporarily saved (to the *volatile* memory - RAM) when you click the **Submit** button on these pages. Parameter settings that are only saved to the device's RAM revert to their previous settings after a hardware/software reset (or power failure). Therefore, to ensure that your configuration changes are retained, you must save them to the device's flash memory using the burn option described below.

➢ **To save the changes to the non-volatile flash memory , take these 2 steps:**

1. Open the 'Maintenance Actions' page (refer to "Maintenance Actions" on page 262).
2. Under the 'Save Configuration' group, click the **BURN** button; a confirmation message appears when the configuration successfully saves.

Notes:

- Saving configuration to the *non-volatile* memory may disrupt traffic on the device. To avoid this, disable all new traffic before saving, by performing a graceful lock (refer to "Locking and Unlocking the Device" on page 264).
- Throughout the Web interface, parameters preceded by the lightning symbol are not applied on-the-fly to the device and require that you reset the device (refer to "Resetting the Device" on page 262) for them to take effect.

3.5.2 Software Update

The **Software Update** menu allows you to upgrade the device's software by loading a new *cmp* file (compressed firmware) along with the *ini* file and a suite of auxiliary files, or to update existing auxiliary files.

The **Software Update** menu includes the following page items:

- Load Auxiliary Files (refer to "Loading Auxiliary Files" on page 266)
- Software Update Key (refer to "Updating the Software Upgrade Key" on page 268)
- Software Upgrade Wizard (refer to "Software Upgrade Wizard" on page 272)
- Configuration File (refer to "Backing Up and Restoring Configuration" on page 276)

3.5.2.1 Loading Auxiliary Files

The 'Load Auxiliary Files' page allows you to load to the device various auxiliary files (described in the table below). For detailed information on these files, refer to "Auxiliary Configuration Files" on page 387. For information on deleting these files from the device, refer to "Device Information" on page 280.

Table 3-64: Auxiliary Files Descriptions

File Type	Description
ini	Provisions the device's parameters. The Web interface enables practically full device provisioning, but customers may occasionally require new feature configuration parameters in which case this file is loaded. **Note:** Loading this file only provisions those parameters that are included in the *ini* file. Parameters that are not specified in the *ini* file are reset to factory default values.
FXS Coefficient	This file contains the telephony interface configuration information for the device. This information includes telephony interface characteristics such as DC and AC impedance, feeding current, and ringing voltage. This file is specific to the type of telephony interface that the device supports. In most cases, you are required to load this type of file. **Note:** This file type is applicable only to Mediant 1000.
CAS	Up to eight different CAS files containing specific CAS protocol definitions for digital modules. These files are provided to support various types of CAS signaling.
Voice Prompts	The voice announcement file contains a set of Voice Prompts (VP) that are played by the device during operation.
Dial Plan	Dial plan file.
Call Progress Tones	This is a region-specific, telephone exchange-dependent file that contains the Call Progress Tones (CPT) levels and frequencies that the device uses. The default CPT file is: U.S.A.
Prerecorded Tones	The *dat* PRT file enhances the device's capabilities of playing a wide range of telephone exchange tones that cannot be defined in the Call Progress Tones file.
User Info	The User Information file maps PBX extensions to IP numbers. This file can be used to represent PBX extensions as IP phones in the global 'IP world'.

➢ To load an auxiliary file to the device using the Web interface, take these 6 steps:

1. Open the 'Load Auxiliary Files' page (**Management** tab > **Software Update** menu > **Load Auxiliary Files** page item).

Figure 3-111: Load Auxiliary Files Page

2. Click the **Browse** button corresponding to the file type that you want to load, navigate to the folder in which the file is located, and then click **Open**; the name and path of the file appear in the field next to the **Browse** button.
3. Click the **Load File** button corresponding to the file you want to load.
4. Repeat steps 2 through 3 for each file you want to load.
5. To save the loaded auxiliary files to flash memory, refer to "Saving Configuration" on page 265.
6. To reset the device (if you have loaded a Call Progress Tones file), refer to "Resetting the Device" on page 262.

Notes:

- Saving an auxiliary file to flash memory may disrupt traffic on the device. To avoid this, disable all traffic on the device by performing a graceful lock (refer to "Locking and Unlocking the Device" on page 264).
- The FXO Coefficient file is applicable only to Mediant 1000.
- You can schedule automatic loading of updated auxiliary files using HTTP, HTTPS, FTP, or NFS (refer to the *Product Reference Manual*).

You can also load the Auxiliary files using the *ini* file. Before you load the files to the device, in the *ini* file you need to include certain *ini* file parameters associated with these files. These *ini* file parameters specify the files that you want loaded and whether they must be stored in the non-volatile memory. For a description of the *ini* file parameters associated with the auxiliary files, refer to "Configuration Files Parameters" on page 383.

➢ To load the auxiliary files via the *ini* file, take these 3 steps:

1. In the *ini* file, define the auxiliary files to be loaded to the device. You can also define in the *ini* file whether the loaded files must be stored in the non-volatile memory so that the TFTP process is not required every time the device boots up.
2. Save the auxiliary files you want to load and the *ini* file in the same directory on your PC.
3. Invoke a BootP/TFTP session; the *ini* and auxiliary files are loaded to the device.

3.5.2.2 Updating the Software Upgrade Key

The device is supplied with a Software Upgrade Key. You can upgrade the device's features, capabilities, and quantity of available resources by by purchasing a new key to match your requirements. The Software Upgrade Key is provided in string format in a text file, which is loaded to the device's non-volatile flash memory. The string defines the device's allowed features and capabilities. A new key overwrites a previously installed key.

You can load a Software Upgrade Key using one of the following:

- Web interface
- BootP/TFTP configuration utility (refer to "Loading via BootP/TFTP" on page 271)
- AudioCodes' EMS (refer to *AudioCodes' EMS User's Manual* or *EMS Product Description*)

Warning: Don't modify the contents of the Software Upgrade Key file.

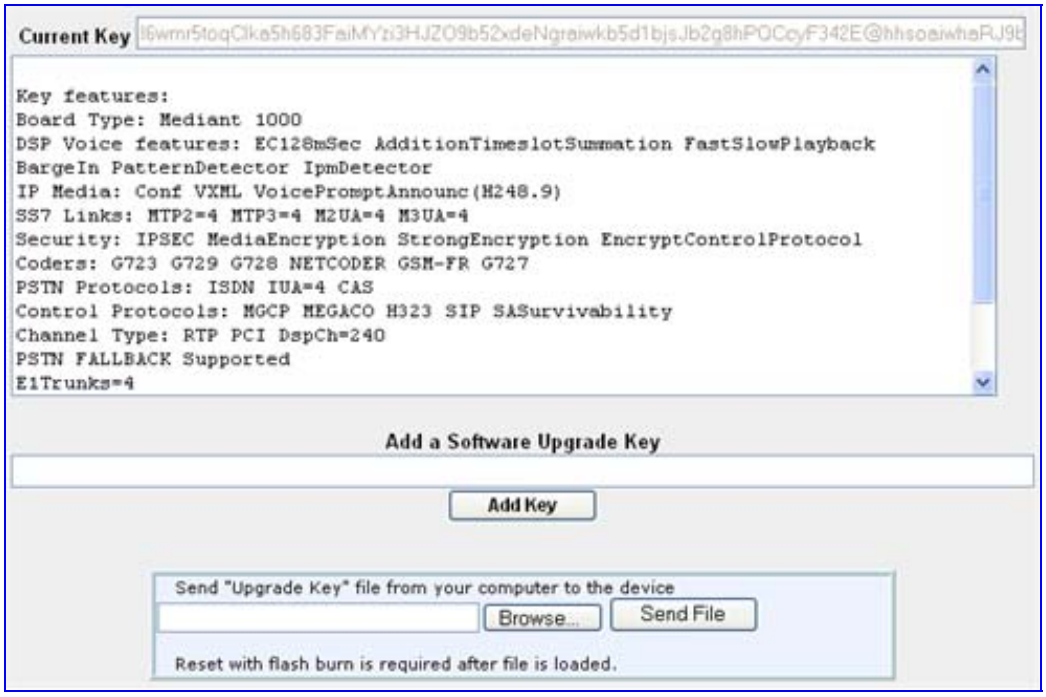
Notes:

- The Software Upgrade Key is an encrypted key.
- The Software Upgrade Key is provided only by AudioCodes.

The procedure below describes how to load a Software Upgrade Key to the device using the Web interface.

➢ To load a Software Upgrade Key, take these 6 steps:

1. Open the 'Software Upgrade Key Status' page (**Management** tab > **Software Update** menu > **Software Update Key** page item).


Current Key
Key features:
Board Type: Mediant 1000
DSP Voice features: EC128mSec AdditionTimeslotSummation FastSlowPlayback
BargeIn PatternDetector IpmDetector
IP Media: Conf VXML VoicePromptAnnounc(H248.9)
SS7 Links: MTP2=4 MTP3=4 M2UA=4 M3UA=4
Security: IPSEC MediaEncryption StrongEncryption EncryptControlProtocol
Coders: G723 G729 G728 NETCODER GSM-FR G727
PSTN Protocols: ISDN IUA=4 CAS
Control Protocols: MGCP MEGACO H323 SIP SASurvivability
Channel Type: RTP PCI DspCh=240
PSTN FALLBACK Supported
E1Trunks=4
Add a Software Upgrade Key
Add Key
Send "Upgrade Key" file from your computer to the device
Browse...
Send File
Reset with flash burn is required after file is loaded.

2. Backup your current Software Upgrade Key as a precaution so that you can re-load this backup key to restore the device's original capabilities if the new key doesn't comply with your requirements:
   a. In the 'Current Key' field, copy the string of text and paste it in any standard text file.
   b. Save the text file to a folder on your PC with a name of your choosing.
3. Open the new Software Upgrade Key file and ensure that the first line displays '[LicenseKeys]' and that it contains one or more lines in the following format: S/N<serial number> = <long Software Upgrade Key>

   For example: S/N370604 = jCx6r5tovCIKaBBbhPtT53Yj...

   One S/N must match the serial number of your device. The device's serial number can be viewed in the 'Device Information' page (refer to "Device Information" on page 280).
4. Follow one of the following procedures, depending on whether you are loading a single or multiple key S/N lines:
   - **Single key S/N line:**
     a. Open the Software Upgrade Key text file (using, for example, Microsoft® Notepad).
     b. Select and copy the key string of the device's S/N and paste it into the field 'Add a Software Upgrade Key'.
     c. Click the **Add Key** button.

- **Multiple S/N lines (as shown below):**

Figure 3-112: Software Upgrade Key with Multiple S/N Lines

a. in the 'Send Upgrade Key file' field, click the **Browse** button and navigate to the folder in which the Software Upgrade Key text file is located on your PC.

b. Click the **Send File** button; the new key is loaded to the device and validated. If the key is valid, it is burned to memory and displayed in the 'Current Key' field.

5. Verify that the Software Upgrade Key file was successfully loaded to the device, by using one of the following methods:

- In the 'Key features' group, ensure that the features and capabilities activated by the installed string match those that were ordered.
- Access the Syslog server (refer to the *Product Reference Manual*) and ensure that the following message appears in the Syslog server: "S/N___ Key Was Updated. The Board Needs to be Reloaded with ini file\n".

6. Reset the device; the new capabilities and resources are active.

Note: If the Syslog server indicates that the Software Upgrade Key file was unsuccessfully loaded (i.e., the 'SN_' line is blank), perform the following preliminary troubleshooting procedures:

1. Open the Software Upgrade Key file and check that the S/N line appears. If it does not appear, contact AudioCodes.
2. Verify that you've loaded the correct file. Open the file and ensure that the first line displays [LicenseKeys].
3. Verify that the contents of the file has not been altered in any way.

3.5.2.2.1 Loading via BootP/TFTP

The procedure below describes how to load a Software Upgrade Key to the device using AudioCodes' BootP/TFTP Server utility (for a detailed description on the BootP utility, refer to the *Product Reference Manual*).

➢ To load a Software Upgrade Key file using BootP/TFTP, take these 6 steps:

1. Place the Software Upgrade Key file (typically, a *.txt file) in the same folder in which the device's *cmp* file is located.
2. Start the BootP/TFTP Server utility.
3. From the **Services** menu, choose **Clients**; the 'Client Configuration' screen is displayed.
4. From the 'INI File' drop-down list, select the Software Upgrade Key file. Note that the device's *cmp* file must be specified in the 'Boot File' field.
5. Configure the initial BootP/TFTP parameters as required, and then click **OK**.
6. Reset the device; the *cmp* and Software Upgrade Key files are loaded to the device.

Note: To load the Software Upgrade Key using BootP/TFTP, the extension name of the key file must be **.ini*.

3.5.2.3 Software Upgrade Wizard

The Software Upgrade Wizard guides you through the process of software upgrade: selecting files and loading them to the device. The wizard also enables you to upgrade software while maintaining the existing configuration. Using the wizard obligates you to load and burn a *cmp* file to the device. You can choose to also use the wizard to load the *ini* and auxiliary files (e.g., Call Progress Tones), but this option cannot be pursued without loading the *cmp* file. For the *ini* and each auxiliary file type, you can choose to reload an existing file, load a new file, or not load a file at all.

The Software Upgrade Wizard allows you to load the following files:

- cmp (mandatory) - compressed firmware file
- *ini* - configuration file
- Auxiliary files:
  - CPT (Call Progress Tone)
  - VP (Voice Prompts)
  - PRT (Prerecorded Tones)
  - CAS
  - FXS Coefficient (applicable only to Mediant 1000)
  - USRINF (User Info)

Warnings:

- Before upgrading the device to a new major software version (e.g., from version 5.2 to 5.4), save a copy of the device's configuration settings (i.e., *ini* file) to your PC (refer to "Backing Up and Restoring Configuration" on page 276), and ensure that you have all the original auxiliary files (e.g., CPT file) currently being used by the device. After you have upgraded the device, upload these files to the device.
- The Software Upgrade Wizard requires the device to be reset at the end of the process, which may disrupt its traffic. To avoid this, disable all traffic on the device before initiating the wizard by performing a graceful lock (refer to "Locking and Unlocking the Device" on page 264).

Notes:

- Before you can load an *ini* or any auxiliary file, you must first load a *cmp* file.
- When you activate the wizard, the rest of the Web interface is unavailable. After you load the desired files, access to the full Web interface is restored.
- You can schedule automatic loading of *cmp, ini,* and auxiliary files using HTTP, HTTPS, FTP, or NFS. (Refer to the *Product Reference Manual*).

➢ To use the Software Upgrade Wizard, take these 11 steps:

1. Stop all traffic on the device (refer to the note above).
2. Open the 'Software Upgrade Wizard' (**Management** tab > **Software Update** menu > **Software Upgrade Wizard**); the 'Software Upgrade Wizard' page appears.

Figure 3-113: Start Software Upgrade Wizard Screen

Software Upgrade Wizard
Start Software Upgrade
Click the button to start the software upgrade process.
Warning:
Before clicking the button 'Start Software Upgrade',
verify that no traffic is running on the device.
Even if you choose to cancel the process in the middle, the device will
reset itself and the previous configuration burned to flash will be reloaded.

3. Click the **Start Software Upgrade** button; the 'Load a CMP file' Wizard page appears.

Figure 3-114: Load a CMP File Wizard Page

Note: At this stage, you can quit the Software Update Wizard, by clicking **Cancel** , without requiring a device reset. However, once you start uploading a cmp file, the process must be completed with a device reset.

4. Click the **Browse** button, navigate to the *cmp* file, and then click **Send File**; the *cmp* file is loaded to the device and you're notified as to a successful loading, as shown below.

Figure 3-115: Loaded CMP File

5. Click one of the following buttons:

- **Reset**; the device resets with the newly loaded *cmp,* and utilizing the current configuration and auxiliary files.
- **Next**; the 'Load an *ini* File' wizard page opens.

Note that as you progress by clicking **Next**, the relevant file name corresponding to the applicable Wizard page is highlighted in the file list on the left.

6. In the 'Load an *ini* File' page, you can now choose to either:

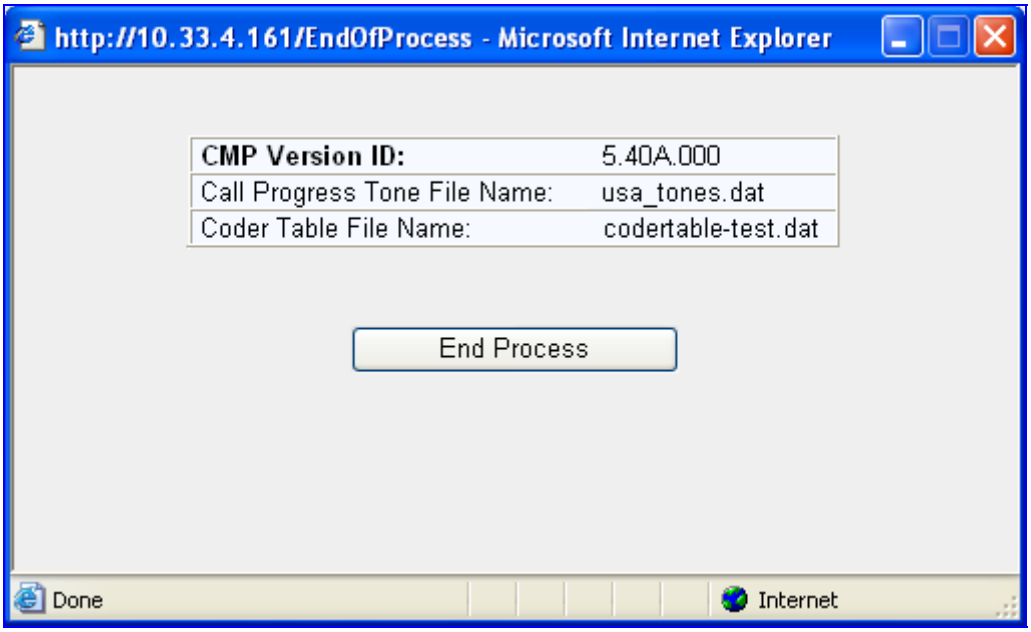
- Click **Browse**, navigate to the *ini* file, and then click **Send File**; the *ini* file is loaded to the device and you're notified as to a successful loading.
- Use the *ini* file currently used by the device, by not selecting an *ini* file and by ensuring that the 'Use existing configuration' check box is marked (default).
- Return the device's configuration settings to factory defaults, by not selecting an *ini* file and by clearing the 'Use existing configuration' check box.

Figure 3-116: Load an ini File Page

7. You can now choose to either:
   - Click **Reset**; the device resets, utilizing the new *cmp* and *ini* file you loaded up to now as well as utilizing the other auxiliary files.
   - Click **Back**; the 'Load a *cmp* file' page is opened again.
   - Click **Next**; the next page opens for loading the next consecutive auxiliary file listed in the Wizard.
8. Follow the same procedure as for loading the *ini* file (Step 6) for loading the auxiliary files.

9. In the 'FINISH' page, complete the upgrade process by clicking **Reset;** the device 'burns' the newly loaded files to flash memory and then resets t.he device. After the device resets, the 'End Process' screen appears displaying the burned configuration files (refer to the figure below).

Figure 3-117: End Process Wizard Page



10. Click **End Process** to close the wizard, and then in the 'Enter Network Password' dialog box, enter your login user name and password (described in "Accessing the Web Interface" on page 26) and click **OK**; a message box appears informing you of the new CMP file:

Figure 3-118: Message Box Informing of Upgraded CMP File

11. Click **OK**; the Web interface now becomes active and reflecting the upgraded device.

3.5.2.4 Backing Up and Restoring Configuration

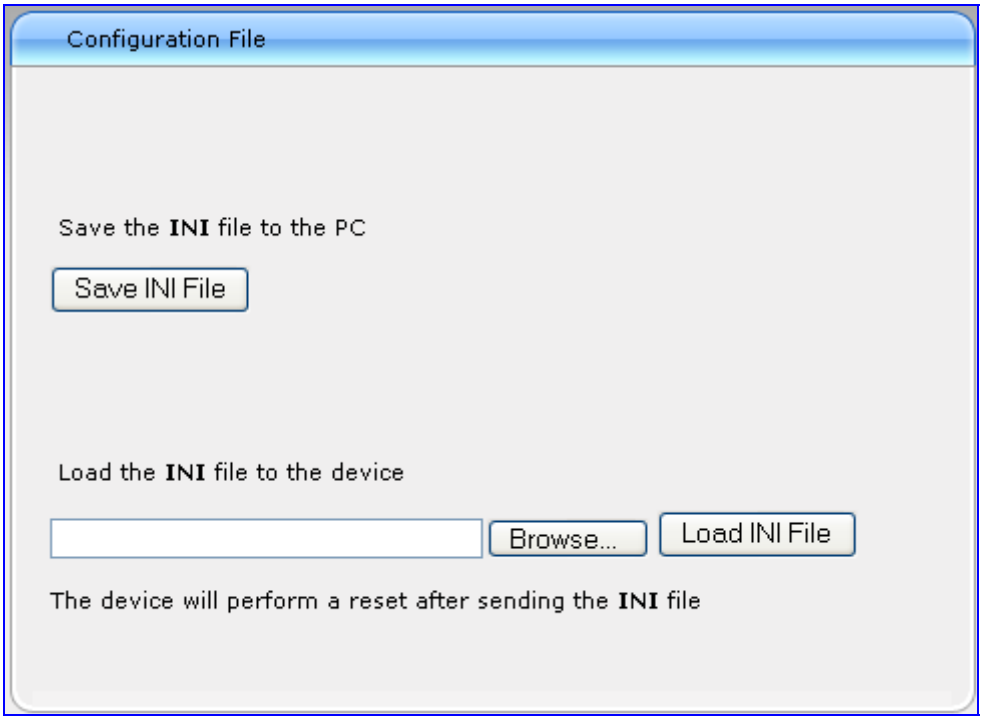
The 'Configuration File' page allows you to save a copy of the device's current configuration file modifications as an *ini* file to a PC. This is useful for backing up your configuration to protect your device configuration. The saved *ini* file includes only those parameters that were modified as well as parameters with other than default values.

In addition, this page allows you to load an *ini* file to the device. If the device has lost its configuration, you can restore the device's configuration by loading the previously saved *ini* file, or by simply loading a newly created *ini* file.

➢ To save and restore the *ini* file, take these 3 steps:

1. Open the 'Configuration File' page (**Management** tab > **Software Update** menu > **Configuration File**).

Figure 3-119: Configuration File Page


Configuration File
Save the INI file to the PC
Save INI File
Load the INI file to the device
Browse...
Load INI File
The device will perform a reset after sending the INI file

2. To save the *ini* file to a PC, perform the following:
   a. Click the **Save INI File** button; the 'File Download' dialog box opens.
   b. Click the **Save** button, navigate to the folder in which you want to save the *ini* file on your PC, and then click **Save**; the device copies the *ini* file to the selected folder.
3. To load an *ini* file to the device, perform the following:
   a. Click the **Browse** button, navigate to the folder in which the *ini* file is located, select the file, and then click **Open**; the name and path of the file appear in the field beside the **Browse** button.
   b. Click the **Load INI File** button, and then at the prompt, click **OK;** the device uploads the *ini* file and then resets (from the *cmp* version stored on the flash memory). Once complete, the 'Enter Network Password' dialog box appears, requesting you to enter your user name and password.

3.6 Status & Diagnostics Tab

The **Status & Diagnostics** tab on the Navigation bar displays all menus related to the operating status of the device and device diagnostics. These menus appear in the Navigation tree and include the following:

- Status & Diagnostics (refer to "Status & Diagnostics" on page 278)
- Gateway Statistics (refer to "Gateway Statistics" on page 283)

3.6.1 Status & Diagnostics

The **Status & Diagnostics** menu is used to view and monitor the device's channels, Syslog messages, hardware and software product information, and to assess the device's statistics and IP connectivity information. This menu includes the following page items:

- Message Log (refer to "Viewing the Device's Syslog Messages" on page 278)
- Ethernet Port Information (refer to "Viewing Ethernet Port Information" on page 280)
- Device Information (refer to "Viewing Device Information" on page 280)
- Performance Statistics (refer to "Viewing Performance Statistics" on page 282)
- Active Alarms (refer to "Viewing Active Alarms" on page 282)

3.6.1.1 Viewing the Device's Syslog Messages

The 'Message Log' page displays Syslog debug messages sent by the device. You can select the Syslog messages in this page, and then copy and paste them into a text editor such as Notepad. This text file (*txt*) can then be sent to AudioCodes Technical Support for diagnosis and troubleshooting.

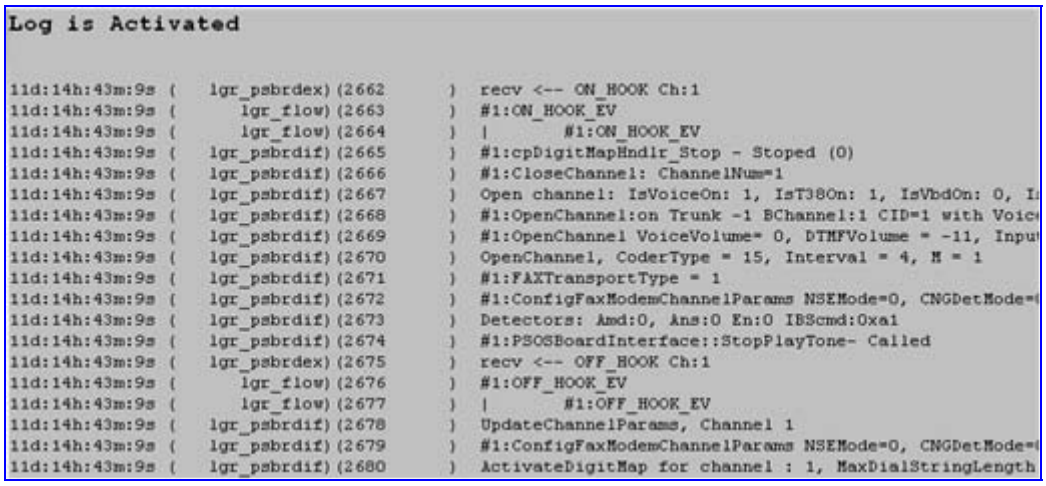
Note: It's not recommended to keep a Message Log session open for a prolonged period. This may cause the device to overload. For prolonged (and detailed) debugging, use an external Syslog server (refer to the *Product Reference Manual*).

➢ **To activate the Message Log, take these 3 steps:**

1. In the 'Advanced Parameters' page (refer "Advanced Parameter" on page 166), set the parameter 'Debug Level' (or *ini* file parameter GwDebugLevel) to 6. This parameter determines the Syslog logging level in the range 0 to 6, where 6 is the highest level.

2. Open the 'Message Log' page (**Status & Diagnostics** tab > **Status & Diagnostics** menu > **Message Log** page item); the 'Message Log' page is displayed and the log is activated.

Figure 3-120: Message Log Screen


Log is Activated
11d:14h:43m:9s (lgr_psbrdex) (2662) recv <-- ON_HOOK Ch:1
11d:14h:43m:9s (lgr_flow) (2663) #1:ON_HOOK_EV
11d:14h:43m:9s (lgr_flow) (2664) | #1:ON_HOOK_EV
11d:14h:43m:9s (lgr_psbrdif) (2665) #1:cpDigitMapHndlr_Stop - Stoped (0)
11d:14h:43m:9s (lgr_psbrdif) (2666) #1:CloseChannel: ChannelNum=1
11d:14h:43m:9s (lgr_psbrdif) (2667) Open channel: IsVoiceOn: 1, IsT38On: 1, IsVbdOn: 0, I
11d:14h:43m:9s (lgr_psbrdif) (2668) #1:OpenChannel:on Trunk -1 BChannel:1 CID=1 with Voic
11d:14h:43m:9s (lgr_psbrdif) (2669) #1:OpenChannel VoiceVolume= 0, DTMFVolume = -11, Inpu
11d:14h:43m:9s (lgr_psbrdif) (2670) OpenChannel, CoderType = 15, Interval = 4, M = 1
11d:14h:43m:9s (lgr_psbrdif) (2671) #1:FAXTransportType = 1
11d:14h:43m:9s (lgr_psbrdif) (2672) #1:ConfigFaxModemChannelParams NSEMode=0, CNGDetMode=
11d:14h:43m:9s (lgr_psbrdif) (2673) Detectors: Amd:0, Ans:0 En:0 IBScmd:0xa1
11d:14h:43m:9s (lgr_psbrdif) (2674) #1:PSOSBoardInterface::StopPlayTone- Called
11d:14h:43m:9s (lgr_psbrdex) (2675) recv <-- OFF_HOOK Ch:1
11d:14h:43m:9s (lgr_flow) (2676) #1:OFF_HOOK_EV
11d:14h:43m:9s (lgr_flow) (2677) | #1:OFF_HOOK_EV
11d:14h:43m:9s (lgr_psbrdif) (2678) UpdateChannelParams, Channel 1
11d:14h:43m:9s (lgr_psbrdif) (2679) #1:ConfigFaxModemChannelParams NSEMode=0, CNGDetMode=
11d:14h:43m:9s (lgr_psbrdif) (2680) ActivateDigitMap for channel : 1, MaxDialStringLength

The displayed logged messages are color coded as follows:

- Yellow - fatal error message
- Blue - recoverable error message (i.e., non-fatal error)
- Black - notice message

3. To clear the page of Syslog messages, in the Navigation tree, click the page item **Message Log** again; the page is cleared and new messages begin appearing.

➢ To stop the Message Log, take this step:

■ Close the page by accessing any another page in the Web interface.

3.6.1.2 Viewing the Ethernet Port Information

The 'Ethernet Port Information' page displays read-only information on the Ethernet connection used by the device. This includes indicating the active port, duplex mode, and speed. You can also access this page from the 'Home' page (refer to "Using the Home Page" on page 52).

For detailed information on the Ethernet redundancy scheme, refer to "Ethernet Interface Redundancy" on page 445. For detailed information on the Ethernet interface configuration, refer to "Ethernet Interface Configuration" on page 445.

➢ To view Ethernet port information, take the following step:

- Open the 'Ethernet Port Information' page (**Status & Diagnostics** tab > **Status & Diagnostics** menu > **Ethernet Port Information** page item).

Figure 3-121: Ethernet Port Information Page

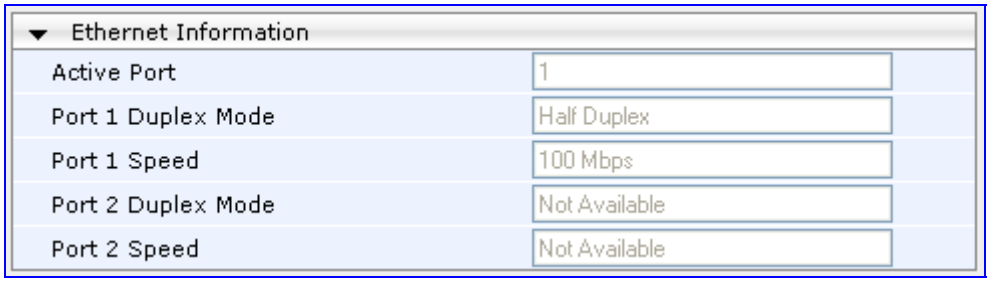


Table 3-65: Ethernet Port Information Parameters

Parameter	Description
Active Port	Displays the active Ethernet port (1 or 2).
Duplex Mode	Displays the Duplex mode of the Ethernet port (Half Duplex or Full Duplex).
Speed	Displays the speed (in Mbps) of the Ethernet port (10 Mbps; 100 Mbps).

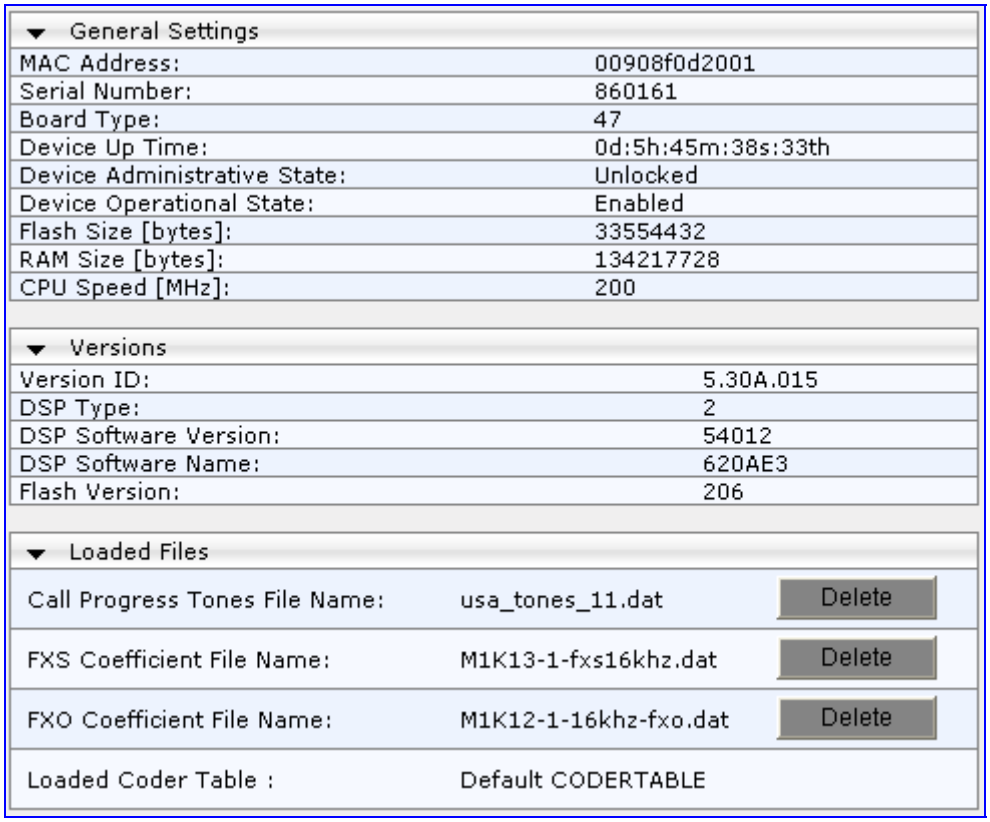
3.6.1.3 Viewing Device Information

The 'Device Information' page displays the device's specific hardware and software product information. This information can help you to expedite troubleshooting. Capture the page and e-mail it to AudioCodes Technical Support personnel to ensure quick diagnosis and effective corrective action. This page also displays any loaded files used by the device (stored in the RAM) and allows you to remove them.

➢ To access the 'Device Information' page, take this step:

- Open the 'Device Information' page (**Status & Diagnostics** tab > **Status & Diagnostics** menu > **Device Information** page item).

Figure 3-122: Device Information Page



➢ To delete any of the loaded files, take this step:

- Click the **Delete** button corresponding to the files that you want to delete. Deleting a file takes effect only after the device is reset (refer to "Resetting the Device" on page 262).

Note: FXS and FXO Coefficient files are applicable only to Mediant 1000.

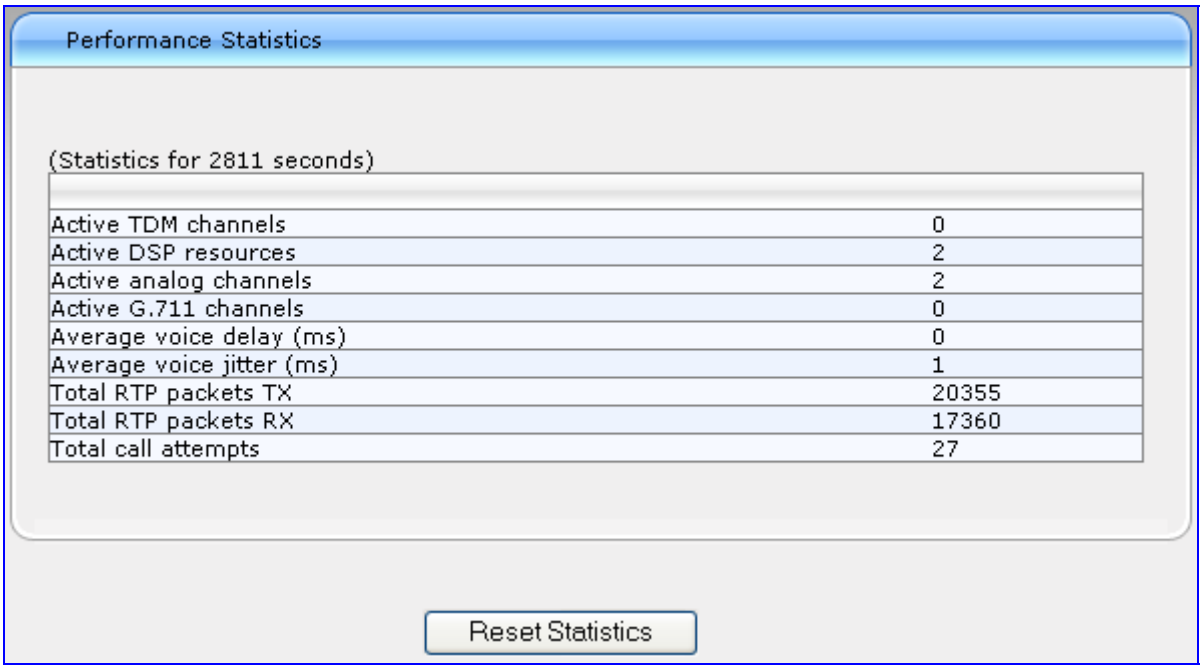
3.6.1.4 Viewing Performance Statistics

The 'Performance Statistics' page provides read-only, device performance statistics. This page is refreshed with new statistics every 60 seconds. The duration that the current statistics has been collected, is displayed above the statistics table.

➢ To view performance statistics, take the following step:

- Open the 'Performance Statistics' page (**Status & Diagnostics** tab > **Status & Diagnostics** menu > **Performance Statistics** page item).

Figure 3-123: Performance Statistics Page


Performance Statistics
(Statistics for 2811 seconds)
Active TDM channels 0
Active DSP resources 2
Active analog channels 2
Active G.711 channels 0
Average voice delay (ms) 0
Average voice jitter (ms) 1
Total RTP packets TX 20355
Total RTP packets RX 17360
Total call attempts 27
Reset Statistics

➢ To reset the performance statistics to zero, take the following step:

- Click the **Reset Statistics** button.

3.6.1.5 Viewing Active Alarms

The 'Active Alarms' page displays a list of currently active alarms. For each alarm, the following information is provided:

- **Severity:** severity level of the alarm:
  - Critical - alarm displayed in red
  - Major - alarm displayed in orange
  - Minor - alarm displayed in yellow
- **Source:** unit from which the alarm was raised
- **Description:** brief explanation of the alarm
- **Date:** date and time that the alarm was generated

You can also access this page from the 'Home' page (refer to "Using the Home Page" on page 52).

➢ To view the list of alarms, take this step:

- Open the 'Active Alarms' page (**Status & Diagnostics** tab > **Status & Diagnostics** menu > **Active Alarms** page item)**.**

Figure 3-124: Active Alarms Page

Severity	Source	Description	Date
Critical	Interface#0/trunk#2	Trunk RAI Alarm.	1.1.2000 , 2:44:20.0
Major	Board#1/EthernetLink#0	Ethernet link alarm. Redundant Link (Physical port #2) is down.	1.1.2000 , 2:3:10.0
Major	Chassis#0/PowerSupply#1	Power-Supply Alarm. Power-Supply is missing	1.1.2000 , 2:3:13.0
Major	Board#1	Board Config Error: psPSTNStopTrunkTraffic User stopped trunk traffic while trunk is runni	1.1.2000 , 2:44:20.0

3.6.2 Gateway Statistics

The 'Gateway Statistics' page allows you to monitor real-time activity such as IP connectivity information, call details and call statistics, including the number of call attempts, failed calls, fax calls, etc. This menu includes the following page items:

- IP to Tel Calls Count and Tel to IP Calls Count (refer to "Call Counters" on page 283)
- Call Routing Status (refer to "Call Routing Status" on page 286)
- SAS/SBC Registered Users (refer to SAS/SBC Registered Users on page 287)
- IP Connectivity (refer to "IP Connectivity" on page 289)

Note: The 'Gateway Statistics' pages don't refresh automatically. To view updated information, re-access the required page.

3.6.2.1 Call Counters

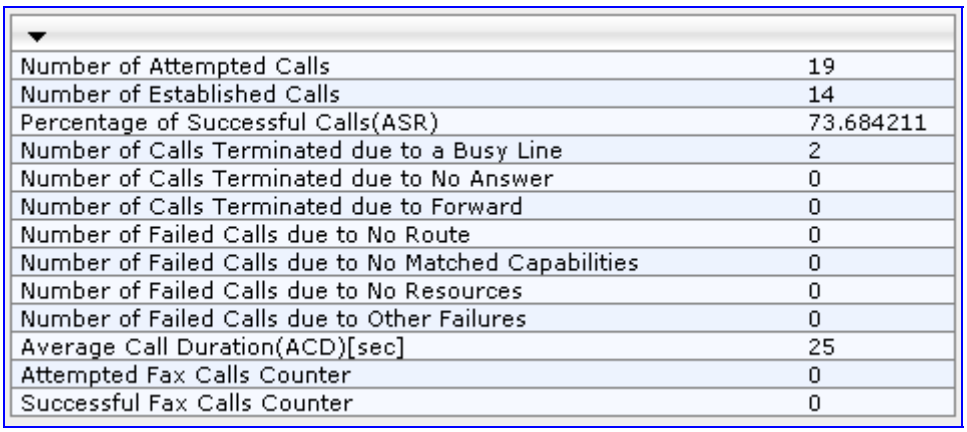
The 'IP to Tel Calls Count' and 'Tel to IP Calls Count' pages provide you with statistical information on incoming (IP-to-Tel) and outgoing (Tel-to-IP) calls. The statistical information is updated according to the release reason that is received after a call is terminated (during the same time as the end-of-call Call Detail Record or CDR message is sent). The release reason can be viewed in the 'Termination Reason' field in the CDR message.

You can reset the statistical data displayed on the page (i.e., refresh the display), by clicking the **Reset Counters** button located on the page.

➢ To view the IP-to-Tel and Tel-to-IP Call Counters pages, take this step:

- Open the Call Counters page that you want to view (**Status & Diagnostics** tab > **Gateway Statistics** menu > **IP to Tel Calls Count** or **Tel to IP Calls Count** page item); the figure below shows the 'IP to Tel Calls Count' page.

Figure 3-125: Calls Count Page



Number of Attempted Calls	19
Number of Established Calls	14
Percentage of Successful Calls(ASR)	73.684211
Number of Calls Terminated due to a Busy Line	2
Number of Calls Terminated due to No Answer	0
Number of Calls Terminated due to Forward	0
Number of Failed Calls due to No Route	0
Number of Failed Calls due to No Matched Capabilities	0
Number of Failed Calls due to No Resources	0
Number of Failed Calls due to Other Failures	0
Average Call Duration(ACD)[sec]	25
Attempted Fax Calls Counter	0
Successful Fax Calls Counter	0

Table 3-66: Call Counters Description

Counter	Description
Number of Attempted Calls	Indicates the number of attempted calls. It is composed of established and failed calls. The number of established calls is represented by the 'Number of Established Calls' counter. The number of failed calls is represented by the failed-call counters. Only one of the established / failed call counters is incremented every time.
Number of Established Calls	Indicates the number of established calls. It is incremented as a result of one of the following release reasons if the duration of the call is greater than zero: ▪ GWAPP_REASON_NOT_RELEVANT (0) ▪ GWAPP_NORMAL_CALL_CLEAR (16) ▪ GWAPP_NORMAL_UNSPECIFIED (31) And the internal reasons: ▪ RELEASE_BECAUSE_UNKNOWN_REASON ▪ RELEASE_BECAUSE_REMOTE_CANCEL_CALL ▪ RELEASE_BECAUSE_MANUAL_DISC ▪ RELEASE_BECAUSE_SILENCE_DISC ▪ RELEASE_BECAUSE_DISCONNECT_CODE **Note:** When the duration of the call is zero, the release reason GWAPP_NORMAL_CALL_CLEAR increments the 'Number of Failed Calls due to No Answer' counter. The rest of the release reasons increment the 'Number of Failed Calls due to Other Failures' counter.
Percentage of Successful Calls (ASR)	The percentage of established calls from attempted calls.
Number of Calls Terminated due to a Busy Line	Indicates the number of calls that failed as a result of a busy line. It is incremented as a result of the following release reason: GWAPP_USER_BUSY (17)

Counter	Description
Number of Calls Terminated due to No Answer	Indicates the number of calls that weren't answered. It's incremented as a result of one of the following release reasons: ▪ GWAPP_NO_USER_RESPONDING (18) ▪ GWAPP_NO_ANSWER_FROM_USER_ALERTED (19) ▪ GWAPP_NORMAL_CALL_CLEAR (16) (when the call duration is zero)
Number of Calls Terminated due to Forward	Indicates the number of calls that were terminated due to a call forward. The counter is incremented as a result of the following release reason: RELEASE_BECAUSE_FORWARD
Number of Failed Calls due to No Route	Indicates the number of calls whose destinations weren't found. It is incremented as a result of one of the following release reasons: ▪ GWAPP_UNASSIGNED_NUMBER (1) ▪ GWAPP_NO_ROUTE_TO_DESTINATION (3)
Number of Failed Calls due to No Matched Capabilities	Indicates the number of calls that failed due to mismatched device capabilities. It is incremented as a result of an internal identification of capability mismatch. This mismatch is reflected to CDR via the value of the parameter DefaultReleaseReason (default is GWAPP_NO_ROUTE_TO_DESTINATION (3)) or by the GWAPP_SERVICE_NOT_IMPLEMENTED_UNSPECIFIED (79) reason.
Number of Failed Calls due to No Resources	Indicates the number of calls that failed due to unavailable resources or a device lock. The counter is incremented as a result of one of the following release reasons: ▪ GWAPP_RESOURCE_UNAVAILABLE_UNSPECIFIED ▪ RELEASE_BECAUSE_GW_LOCKED
Number of Failed Calls due to Other Failures	This counter is incremented as a result of calls that failed due to reasons not covered by the other counters.
Average Call Duration (ACD) [sec]	The average call duration (ACD) in seconds of established calls. The ACD value is refreshed every 15 minutes and therefore, this value reflects the average duration of all established calls made within a 15 minute period.
Attempted Fax Calls Counter	Indicates the number of attempted fax calls.
Successful Fax Calls Counter	Indicates the number of successful fax calls.

3.6.2.2 Call Routing Status

The 'Call Routing Status' page provides you with information on the current routing method used by the device. This information includes the IP address and FQDN (if used) of the Proxy server with which the device currently operates.

➢ To view the call routing status, take this step:

- Open the 'Call Routing Status' page (**Status & Diagnostics** tab > **Gateway Statistics** menu > **Calls Routing Status** page item).

Figure 3-126: Call Routing Status Page

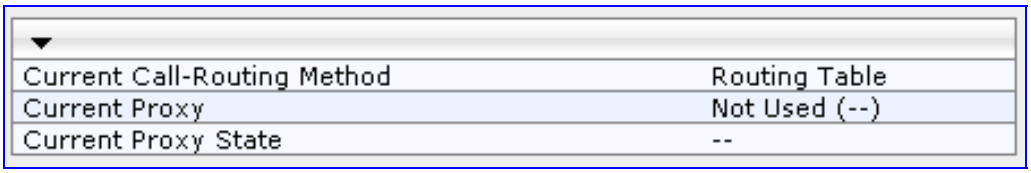


Table 3-67: Call Routing Status Parameters

Parameter	Description
Current Call-Routing Method	▪ Proxy = Proxy server is used to route calls. ▪ Routing Table preferred to Proxy = The 'Tel to IP Routing' table takes precedence over a Proxy for routing calls ('Prefer Routing Table' parameter is set to 'Yes' as described in "Proxy & Registration Parameters" on page 146).
Current Proxy	▪ Not Used = Proxy server isn't defined. ▪ IP address and FQDN (if exists) of the Proxy server with which the device currently operates.
Current Proxy State	▪ N/A = Proxy server isn't defined. ▪ OK = Communication with the Proxy server is in order. ▪ Fail = No response from any of the defined Proxies.

3.6.2.3 Registration Status

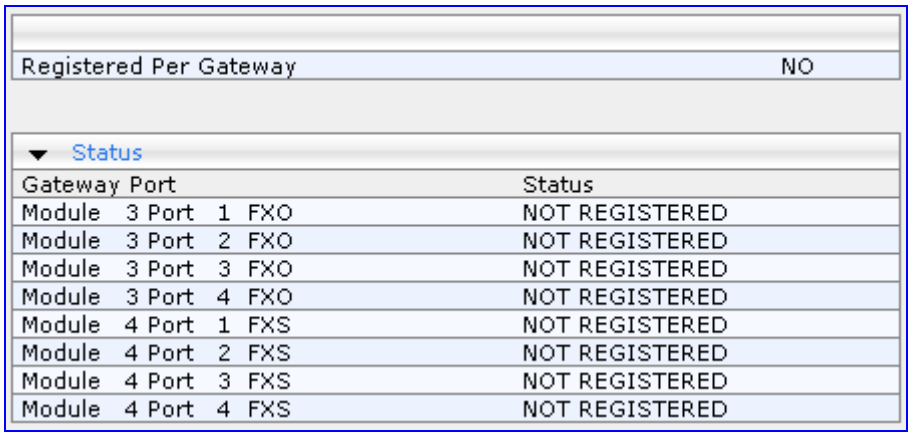
The 'Registration Status' page displays whether the device or endpoints are registered to a SIP Registrar/Proxy server.

Note: The 'Registration Status' page is applicable only to Mediant 1000.

➢ To view endpoint Registration status, take this step:

- Open the 'Registration Status' page (**Status & Diagnostics** tab > **Gateway Statistics** menu > **Registration Status** page item).

Figure 3-127: Registration Status Page


Registered Per Gateway NO
Status
Gateway Port Status
Module 3 Port 1 FXO NOT REGISTERED
Module 3 Port 2 FXO NOT REGISTERED
Module 3 Port 3 FXO NOT REGISTERED
Module 3 Port 4 FXO NOT REGISTERED
Module 4 Port 1 FXS NOT REGISTERED
Module 4 Port 2 FXS NOT REGISTERED
Module 4 Port 3 FXS NOT REGISTERED
Module 4 Port 4 FXS NOT REGISTERED

If a channel is registered, then 'REGISTERED' is displayed in the 'Status' column corresponding to the channel; otherwise, 'NOT REGISTERED' is displayed. If registration is per device, then 'YES' is displayed alongside 'Registered Per Gateway'; otherwise, 'NO' is displayed.

3.6.2.4 SAS/SBC Registered Users

The 'SAS Registered Users' page displays a list of up to 100 Stand Alone Survivability (SAS) registered users. The SAS feature is configured in the 'SAS Configuration' page (refer to "Stand-Alone Survivability" on page 188).

Note: The SAS feature is applicable only to Mediant 1000.

➢ **To view the SAS registered users, take this step:**

- Open the 'SAS Registered Users' page (**Status & Diagnostics** tab > **Gateway Statistics** menu > **SAS/SBC Registered Users** page item).

Figure 3-128: SAS Registered Users Page

Address Of Record	Contact
<sip:2400@Proxies.ac>	<sip:2400@10.8.210.5>;expires=180
<sip:2401@Proxies.ac>	<sip:2401@10.8.210.5>;expires=180
<sip:2500@Proxies.ac>	<sip:2500@10.8.210.5>;expires=180
<sip:2402@Proxies.ac>	<sip:2402@10.8.210.5>;expires=180
<sip:2403@Proxies.ac>	<sip:2403@10.8.210.5>;expires=180
<sip:2404@Proxies.ac>	<sip:2404@10.8.210.5>;expires=180
<sip:2405@Proxies.ac>	<sip:2405@10.8.210.5>;expires=180

Table 3-68: SAS Registered Users Parameters

Column Name	Description
Address of Record	An address-of-record (AOR) is a SIP or SIPS URI that points to a domain with a location service that can map the URI to another URI (Contact) where the user might be available.
Contact	SIP URI that can be used to contact that specific instance of the User Agent for subsequent requests.

3.6.2.5 IP Connectivity

The 'IP Connectivity' page displays online, read-only network diagnostic connectivity information on all destination IP addresses configured in the 'Tel to IP Routing' page (refer to "Tel to IP Routing Table" on page 200).

Notes:

- This information is available only if the parameter 'Enable Alt Routing Tel to IP' (refer to "Routing General Parameters" on page 198) is set to 1 (Enable) or 2 (Status Only).
- The information in columns 'Quality Status' and 'Quality Info' (per IP address) is reset if two minutes elapse without a call to that destination.

➢ **To view the IP connectivity information, take these 2 steps:**

1. In the 'Routing General Parameters' page, set the parameter 'Enable Alt Routing Tel to IP' (or *ini* file parameter AltRoutingTel2IPEnable) to Enable **[1]** or Status Only **[2]**.
2. Open the 'IP Connectivity' page (**Status & Diagnostics** tab > **Gateway Statistics** menu > **IP Connectivity** page item).

Figure 3-129: IP Connectivity Page

	IP Address	Host Name	Connectivity Method	Connectivity Status	Quality Status	Quality Info	DNS Status
1	Unused	---	---	---	---	---	---
2	Unused	---	---	---	---	---	---
3	Unused	---	---	---	---	---	---
4	Unused	---	---	---	---	---	---
5	Unused	---	---	---	---	---	---
6	Unused	---	---	---	---	---	---
7	Unused	---	---	---	---	---	---
8	Unused	---	---	---	---	---	---
9	Unused	---	---	---	---	---	---
10	Unused	---	---	---	---	---	---

Table 3-69: IP Connectivity Parameters

Column Name	Description
IP Address	The IP address can be one of the following: ▪ IP address defined as the destination IP address in the 'Tel to IP Routing' table (refer to "Tel to IP Routing Table" on page 200). ▪ IP address resolved from the host name defined as the destination IP address in the 'Tel to IP Routing' table.
Host Name	Host name (or IP address) defined as the destination IP address in the 'Tel to IP Routing' table.
Connectivity Method	The method according to which the destination IP address is queried periodically (ICMP ping or SIP OPTIONS request).

Column Name	Description
Connectivity Status	The status of the IP address' connectivity according to the method in the 'Connectivity Method' field. ▪ OK = Remote side responds to periodic connectivity queries. ▪ Lost = Remote side didn't respond for a short period. ▪ Fail = Remote side doesn't respond. ▪ Init = Connectivity queries not started (e.g., IP address not resolved). ▪ Disable = The connectivity option is disabled, i.e., parameter 'Alt Routing Tel to IP Mode' (AltRoutingTel2IPMode *ini*) is set to 'None' or 'QoS' (refer to "Routing General Parameters" on page 198).
Quality Status	Determines the QoS (according to packet loss and delay) of the IP address. ▪ Unknown = Recent quality information isn't available. ▪ OK ▪ Poor **Notes:** ▪ This parameter is applicable only if the parameter 'Alt Routing Tel to IP Mode' is set to 'QoS' or 'Both' (AltRoutingTel2IPMode = 2 or 3). ▪ This parameter is reset if no QoS information is received for 2 minutes.
Quality Info.	Displays QoS information: delay and packet loss, calculated according to previous calls. **Notes:** ▪ This parameter is applicable only if the parameter 'Alt Routing Tel to IP Mode' is set to 'QoS' or 'Both' (AltRoutingTel2IPMode = 2 or 3). ▪ This parameter is reset if no QoS information is received for 2 minutes.
DNS Status	DNS status can be one of the following: ▪ DNS Disable ▪ DNS Resolved ▪ DNS Unresolved

4 ini File Configuration

As an alternative to configuring the device using the Web interface (as described in "Web-Based Management" on page 25), you can configure the device by loading an *ini* file containing user-defined parameters. The *ini* file can be loaded using the following methods:

- AudioCodes' BootP/TFTP utility (refer to the *Product Reference Manual*)
- Any standard TFTP server
- Web interface (refer to "Backing Up and Restoring Configuration" on page 276)

The *ini* file configuration parameters are saved in the device's non-volatile memory after the file is loaded to the device. When a parameter is absent from the *ini* file, the default value is assigned to that parameter (according to the *cmp* file loaded to the device) and stored in the non-volatile memory (thereby, overriding the value previously defined for that parameter).

Some of the device's parameters are configurable only through the *ini* file (and not the Web interface). These parameters usually determine a low-level functionality and are seldom changed for a specific application.

Notes:

- For a list of the *ini* file parameters, refer to "The ini File Parameter Reference" on page 296. The *ini* file parameters that are configurable in the Web interface are described in "Web-Based Management" on page 25. The *ini* parameters that can't be configured using the Web interface are described in this section.
- To define or restore default settings using the *ini* file, refer to "Default Settings" on page 385.

4.1 Secured Encoded ini File

The *ini* file contains sensitive information that is required for the functioning of the device. Typically, it is loaded to or retrieved from the device using TFTP or HTTP. These protocols are not secure and vulnerable to potential hackers.

To overcome this security threat, the AudioCodes' TrunkPack Downloadable Conversion Utility (DConvert) allows you to binary-encode the *ini* file before loading it to the device (refer to the *Product Reference Manual)*. If you retrieve an *ini* file from the device using the Web interface (refer to Backing Up and Restoring Configuration) that was initially loaded as encoded to the device, the file is retrieved as encoded and vice versa.

Note: The procedure for loading an encoded *ini* file is identical to the procedure for loading an unencoded *ini* file.

4.2 The ini File Structure

The *ini* file can contain any number of parameters. The *ini* file can contain the following types of parameters:

- Individual parameters, which are conveniently grouped (optional) by their functionality (refer to "Structure of Individual ini File Parameters" on page 292)
- Table parameters, which include multiple individual parameters (refer to "Structure of ini File Table Parameters" on page 293)

4.2.1 Structure Rules

The *ini* file must adhere to the following format rules:

- The *ini* file name must not include hyphens (-) or spaces; if necessary, use an underscore (_) instead.
- Lines beginning with a semi-colon (;) are ignored. These can be used for adding remarks in the *ini* file.
- A carriage return (i.e., Enter) must be done at the end of each line.
- The number of spaces before and after the equals sign (=) is irrelevant.
- Subsection names for grouping parameters are optional.
- If there is a syntax error in the parameter name, the value is ignored.
- Syntax errors in the parameter's value can cause unexpected errors (parameters may be set to the incorrect values).
- Parameter string values that denote file names (e.g., CallProgressTonesFileName), must be enclosed with inverted commas ('…'), e.g., CallProgressTonesFileName = 'cpt_usa.dat'
- The parameter name is not case-sensitive.
- The parameter value is not case-sensitive, except for coder names.
- The *ini* file must end with at least one carriage return.

4.2.2 Structure of Individual ini File Parameters

The structure of individual *ini* file parameters in an *ini* file is shown below:

```
[Subsection Name]
Parameter Name = Parameter Value
Parameter_Name = Parameter_Value
; REMARK
```

An example of an *ini* file containing individual *ini* file parameters is shown below:

```
[SYSTEM Params]
SyslogServerIP = 10.13.2.69
EnableSyslog = 1
; These are a few of the system-related parameters.
[WEB Params]
LogoWidth = '339'
WebLogoText = 'My Device'
UseWeblogo = 1
; These are a few of the Web-related parameters.
[Files]
CallProgressTonesFileName = 'cpusa.dat'
```

4.2.3 Structure of ini File Table Parameters

You can use an*ini* file to configure table parameters, which include several parameters (table *columns*) grouped according to the applications they configure (e.g., NFS and IPSec). When loading an *ini* file to the device, it's recommended to include only tables that belong to applications that are to be configured (dynamic tables of other applications are empty, but static tables are not).

A table is defined as a *secret* table (i.e., concealed) if it contains at least one secret data field or if it depends on another secret table. For example, in the IPSec application, IPSec tables are defined as secret tables as the IKE table contains a pre-shared key that must be concealed. Therefore, the SPD table that depends on the IKE table is defined as a secret table as well. Secret tables are always concealed when loading an *ini* file to the device. However, there is a commented title that states that the secret table exists in the device, but is not to be revealed. Secret tables are always stored in the device's non-volatile memory and can be overwritten by new tables that are provided in a new *ini* file. If a secret table appears in an *ini* file, it replaces the current table regardless of its content. To delete a secret table from the device, include an empty table of the same type (with no data lines) as part of a new *ini* file.

The *ini* file table parameter is composed of the following elements:

- **Title of the table:** The name of the table in square brackets (e.g., [MY_TABLE_NAME]).
- **Format line:** Specifies the columns (parameters) of the table (by their string names) that are to be configured.
  - The first word of the Format line must be 'FORMAT', followed by the Index field name, and then an equal (=) sign. After the equal sign, the names of the parameters (*items*) are listed.
  - Items must be separated by a comma (,).
  - The Format line must only include columns that can be modified (i.e., parameters that are not specified as read-only). An exception is Index fields that are always mandatory.
  - The Format line must end with a semicolon (;).

- **Data line(s):** Contain the actual values of the parameters. The values are interpreted according to the Format line.
  - The first word of the Data line must be the table's string name followed by the Index field.
  - Items must be separated by a comma (,).
  - A Data line must end with a semicolon (;).
- **End-of-Table Mark:** Indicates the end of the table. The same string used for the table's title, preceded by a backslash (\), e.g., [\MY_TABLE_NAME].

The following displays an example of the structure of an *ini* file table parameter.

```
[Table_Title]
; This is the title of the table.
FORMAT Item Index = Item Name1, Item Name2, Item Name3;
; This is the Format line.
Item 0 = value1, value2, value3;
Item 1 = value1, $$, value3;
; These are the Data lines.
[\Table Title]
; This is the end-of-the-table-mark.
```

Refer to the following notes:

- Indices (in both the Format and the Data lines) must appear in the same order. The Index field must never be omitted.
- The Format line can include a subset of the configurable fields in a table. In this case, all other fields are assigned with the pre-defined default values for each configured line.
- The order of the fields in the Format line isn't significant (as opposed to the Index fields). The fields in the Data lines are interpreted according to the order specified in the Format line.
- The double dollar sign ($$) in a Data line indicates the default value for the parameter.
- The order of the Data lines is insignificant.
- Data lines must match the Format line, i.e., it must contain exactly the same number of Indices and Data fields and must be in exactly the same order.
- A line in a table is identified by its table name and Index fields. Each such line may appear only once in the *ini* file.
- Table dependencies: Certain tables may depend on other tables. For example, one table may include a field that specifies an entry in another table. This method is used to specify additional attributes of an entity, or to specify that a given entity is part of a larger entity. The tables must appear in the order of their dependency (i.e., if Table X is referred to by Table Y, Table X must appear in the *ini* file before Table Y).

The table below displays an example of an *ini* file table parameter:

```
[ PREFIX ]
FORMAT PREFIX Index = PREFIX DestinationPrefix,
PREFIX DestAddress, PREFIX SourcePrefix, PREFIX ProfileId,
PREFIX_MeteringCode, PREFIX_DestPort;
PREFIX 0 = 10, 10.13.83.5, *, 0, 255, 0;
PREFIX 1 = 20, 10.13.83.7, *, 0, 255, 0;
PREFIX 2 = 30, 10.13.83.6, *, 0, 255, 0;
PREFIX 3 = 20, 10.13.83.2, *, 0, 255, 0;
[ \PREFIX ]
```

Note: Do not include read-only parameters in the *ini* file table parameter, as this can cause an error when trying to load the file to the device.

4.2.4 Example of an ini File

Below is an example of an *ini* file for the VoIP device.

```
;Channel Params
DJBufMinDelay = 75
RTPRedundancyDepth = 1
IsProxyUsed = 1
ProxyIP = 192.168.122.179
[CoderName]
FORMAT CoderName Index = CoderName Type, CoderName PacketInterval,
CoderName_rate, CoderName_PayloadType, CoderName_Sce;
CoderName 1= g7231,90
[\CoderName]
;List of serial B-channel numbers
[TrunkGroup]
FORMAT TrunkGroup Index = TrunkGroup TrunkGroupNum,
TrunkGroup_FirstTrunkId,TrunkGroup_LastTrunkId,
TrunkGroup FirstBChannel, TrunkGroup LastBChannel,
TrunkGroup FirstPhoneNumber, TrunkGroup ProfileId,
TrunkGroup_Module;
TrunkGroup 1 = 0,0,0,1,24,1000;
TrunkGroup 2 = 0,1,1,1,24,2000;
TrunkGroup 3 = 0,2,2,1,24,3000;
TrunkGroup 4 = 0,3,3,1,24,4000;
[\TrunkGroup]
CallProgressTonesFilename = 'CPUSA.dat'
SaveConfiguration = 1
```

4.3 Modifying an ini File

You can modify an *ini* file currently used by a device. Modifying an *ini* file instead of loading an entirely new *ini* file preserves the device's current configuration, including factory default values.

➢ To modify an *ini* file, take these 4 steps:

1. Save the *ini* file from the device to your PC using the Web interface (refer to "Backing Up and Restoring Configuration" on page 276).
2. Open the *ini* file (using a text file editor such as Microsoft Notepad), and then modify the *ini* file parameters according to your requirements.
3. Save the modified *ini* file, and then close the file.
4. Load the modified *ini* file to the device, using either the BootP/TFTP utility or the Web interface (refer to "Backing Up and Restoring Configuration" on page 276).

Tip: Before loading the *ini* file to the device, verify that the file extension of the *ini* file saved on your PC is correct, i.e., **.ini*.

4.4 Reference for ini File Parameters

This subsection lists all the *ini* file parameters. References to their descriptions in the Web interface are provided except for those *ini* file parameters that can only be configured using the *ini* file.

4.4.1 Networking Parameters

Table 4-1: Networking Parameters

ini File Parameter	Description
EthernetPhyConfiguration	Defines the Ethernet connection mode type. ▪ **[0]** = 10Base-T half-duplex ▪ **[1]** = 10Base-T full-duplex ▪ **[2]** = 100Base-TX half-duplex ▪ **[3]** = 100Base-TX full-duplex ▪ **[4]** = Auto-negotiate (default) For detailed information on Ethernet interface configuration, refer to "Ethernet Interface Configuration" on page 445.

ini File Parameter	Description
MllRedundancyEnable	Enables the Ethernet Interface Redundancy feature. When enabled, the device performs a switchover to the secondary (redundant) Ethernet port upon sensing a link failure in the primary Ethernet port. When disabled, the device operates with a single port (i.e. no redundancy support). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable For detailed information on Ethernet interface redundancy, refer to Ethernet Interface Redundancy on page 445. **Note:** For this parameter to take effect, a device reset is required.
DHCPEnable	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.
DHCPSpeedFactor	Determines the DHCP renewal speed. ▪ **[0]** = Disable ▪ **[1]** = Normal (default) ▪ **[2]** to **[10]** = Fast When set to 0, the DHCP lease renewal is disabled. Otherwise, the renewal time is divided by this factor. Some DHCP-enabled routers perform better when set to 4.
EnableDHCPLeaseRenewal	Enables or disables DHCP renewal support. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable This parameter is applicable only if DHCPEnable is set to 0 for cases where booting up the device via DHCP is not desirable, but renewing DHCP leasing is. When the device is powered up, it attempts to communicate with a BootP server. If there is no response and if DHCP is disabled, the device boots from flash. It then attempts to communicate with the DHCP server to renew the lease.
EnableLANWatchDog	For a description of this parameter, refer to "General Parameters" on page 166.
DNSPriServerIP	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.
DNSSecServerIP	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.

ini File Parameter	Description
DNS2IP	This *ini* file table parameter configures the internal DNS table for resolving host names to IP addresses. Two different IP addresses (in dotted-decimal notation) can be assigned to a host name. The format of this parameter is as follows: [Dns2Ip] FORMAT Dns2Ip_**Index** = Dns2Ip_**DomainName**, Dns2Ip_**FirstIpAddress**, Dns2Ip_**SecondIpAddress**; [\Dns2Ip] Where, ▪ DomainName = Host name. ▪ FirstIpAddress = First IP address. ▪ SecondIpAddress = Second IP address. For example: [Dns2Ip] Dns2Ip 0 = DnsName, 1.1.1.1, 2.2.2.2; [\Dns2Ip] **Notes:** ▪ If the internal DNS table is used, the device first attempts to resolve a domain name using this table. If the domain name isn't found, the device performs a DNS resolution using an external DNS server. ▪ This parameter can include up to 10 indices. ▪ To configure the internal DNS table using the Web interface, refer to "Internal DNS Table" on page 207. ▪ For an explanation on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
SRV2IP	This *ini* file table parameter defines the internal SRV table for resolving host names to DNS A-Records. Three different A-Records can be assigned to a host name. Each A-Record contains the host name, priority, weight, and port. The format of this parameter is as follows: [SRV2IP] FORMAT SRV2IP_**Index** = SRV2IP_**InternalDomain**, SRV2IP_**TransportType**, SRV2IP_**Dns1**, SRV2IP_**Priority1**, SRV2IP_**Weight1**, SRV2IP_**Port1**, SRV2IP_**Dns2**, SRV2IP_**Priority2**, SRV2IP_**Weight2**, SRV2IP_**Port2**, SRV2IP_**Dns3**, SRV2IP_**Priority3**, SRV2IP_**Weight3**, SRV2IP_**Port3**; [\SRV2IP] Where, ▪ InternalDomain = Internal domain name. ▪ TransportType = Transport type. ▪ Dns1, Dns2, Dns3 = DNS name 1, 2, and 3. ▪ Priority1, Priority2, Priority3 = Priority 1, 2, and 3. ▪ Weight1, Weight2, Weight3 = Weight 1, 2, and 3. ▪ Port1, Port2, Port3 = Port 1, 2, and 3. For example: [SRV2IP] SRV2IP 0 = SrvDomain,0,Dnsname1,1,1,500,Dnsname2,2,2,501,$$,0,0,0; [\SRV2IP]

ini File Parameter	Description
	Notes: ▪ If the Internal SRV table is used, the device first attempts to resolve a domain name using this table. If the domain name isn't located, the device performs an SRV resolution using an external DNS server. ▪ This parameter can include up to 10 indices. ▪ To configure the Internal SRV table using the Web interface, refer to "Internal SRV Table" on page 208. ▪ For an explanation on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
EnableSTUN	For a description of this parameter, refer to "Configuring the Application Settings" on page 71.
STUNServerPrimaryIP	For a description of this parameter, refer to "Configuring the Application Settings" on page 71.
STUNServerSecondaryIP	For a description of this parameter, refer to "Configuring the Application Settings" on page 71.
STUNServerDomainName	Defines the domain name for the Simple Traversal of User Datagram Protocol (STUN) server's address (used for retrieving all STUN servers with an SRV query). The STUN client can perform the required SRV query to resolve this domain name to an IP address and port, sort the server list, and use the servers according to the sorted list. **Note:** Use either the STUNServerPrimaryIP or the STUNServerDomainName parameter, with priority to the first one.
NATBindingDefaultTimeout	Defines the default NAT binding lifetime in seconds. STUN is used to refresh the binding information after this time expires. The valid range is 0 to 2,592,000. The default value is 30.
DisableNAT	Enables / disables the Network Address Translation (NAT) mechanism. ▪ **[0]** = Enabled. ▪ **[1]** = Disabled (default). **Note:** The compare operation that is performed on the IP address is enabled by default and is controlled by the parameter EnableIPAddrTranslation. The compare operation that is performed on the UDP port is disabled by default and is controlled by the parameter EnableUDPPortTranslation.
EnableIPAddrTranslation	▪ **[0]** = Disable IP address translation. ▪ **[1]** = Enable IP address translation for RTP, RTCP and T.38 packets (default). ▪ **[2]** = Enable IP address translation for ThroughPacket™. ▪ **[3]** = Enable IP address translation for all protocols (RTP, RTCP, T38 and ThroughPacket™). When enabled, the device compares the source IP address of the first incoming packet, to the remote IP address stated in the opening of the channel. If the two IP addresses don't match, the NAT mechanism is activated. Consequently, the remote IP address of the outgoing stream is replaced by the source IP address of the first incoming packet. **Note:** The NAT mechanism must be enabled for this parameter to take effect (DisableNAT = 0).

ini File Parameter	Description
EnableUDPPortTranslation	▪ **[0]** = Disable UDP port translation (default). ▪ **[1]** = Enable UDP port translation. When enabled, the device compares the source UDP port of the first incoming packet, to the remote UDP port stated in the opening of the channel. If the two UDP ports don't match, the NAT mechanism is activated. Consequently, the remote UDP port of the outgoing stream is replaced by the source UDP port of the first incoming packet. **Note:** The NAT mechanism and the IP address translation must be enabled for this parameter to take effect (DisableNAT = 0, EnableIpAddrTranslation = 1).
NoOpEnable	Enables or disables the transmission of RTP or T.38 No-Op packets. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable This mechanism ensures that the NAT binding remains open during RTP or T.38 silence periods.
NoOpInterval	Defines the time interval in which RTP or T.38 No-Op packets are sent in the case of silence (no RTP / T.38 traffic) when No-Op packet transmission is enabled. The valid range is 20 to 65,000 msec. The default is 10,000. **Note:** To enable No-Op packet transmission, use the NoOpEnable parameter.
RTPNoOpPayloadType	Determines the payload type of No-Op packets. The valid range is 96 to 127 (for the range of Dynamic RTP Payload Type for all types of non hard-coded RTP Payload types, refer to RFC 3551). The default value is 120. **Note:** When defining this parameter, ensure that it doesn't cause collision with other payload types.
EnableDetectRemoteMACChange	Changes the RTP packets according to the MAC address of received RTP packets and according to Gratuitous Address Resolution Protocol (GARP) messages. ▪ **[0]** = nothing is changed. ▪ **[1]** = If the device receives RTP packets with a different source MAC address (than the MAC address of the transmitted RTP packets), then it sends RTP packets to this MAC address and removes this IP entry from the device's ARP cache table. ▪ **[2]** = The device uses the received GARP packets to change the MAC address of the transmitted RTP packets. ▪ **[3]** = both 1 and 2 options above are used (default).
StaticNatIP	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.
SyslogServerIP	For a description of this parameter, refer to "Configuring the Management Settings" on page 254.
SyslogServerPort	For a description of this parameter, refer to "Configuring the Management Settings" on page 254.
EnableSyslog	For a description of this parameter, refer to "Configuring the Management Settings" on page 254.

ini File Parameter	Description
SyslogOutputMethod	Determines the method used for Syslog messages. ▪ **[0]** = Send all Syslog messages to the defined Syslog server (default). ▪ **[1]** = Send all Syslog messages using the Debug Recording mechanism. ▪ **[2]** = Send only Error and Warning level Syslog messages using the Debug Recording mechanism. For a detailed description on Debug Recording, refer to Debug Recording (DR).
BaseUDPport	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88.
RemoteBaseUDPPort	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88.
L1L1ComplexTxUDPPort	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88.
L1L1ComplexRxUDPPort	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88.
NTPServerIP	For a description of this parameter, refer to "Configuring the Application Settings" on page 71.
NTPServerUTCOffset	For a description of this parameter, refer to "Configuring the Application Settings" on page 71.
NTPUpdateInterval	For a description of this parameter, refer to "Configuring the Application Settings" on page 71.
IP Routing Table parameters: The IP routing *ini* file parameters are array parameters. Each parameter configures a specific column in the IP routing table. The first entry in each parameter refers to the first row in the IP routing table, the second entry to the second row and so forth. In the following example, two rows are configured when the device is in network 10.31.x.x: RoutingTableDestinationsColumn = 130.33.4.6, 83.4.87.6 RoutingTableDestinationMasksColumn = 255.255.255.255, 255.255.255.0 RoutingTableGatewaysColumn = 10.31.0.1, 10.31.0.112 RoutingTableInterfacesColumn = 0, 1 RoutingTableHopsCountColumn = 20, 20	
RoutingTableDestinationsColumn	For a description of this parameter, refer to "Configuring the IP Routing Table" on page 75.
RoutingTableDestinationMasksColumn	For a description of this parameter, refer to "Configuring the IP Routing Table" on page 75.
RoutingTableGatewaysColumn	For a description of this parameter, refer to "Configuring the IP Routing Table" on page 75.
RoutingTableHopsCountColumn	For a description of this parameter, refer to "Configuring the IP Routing Table" on page 75.
RoutingTableInterfacesColumn	For a description of this parameter, refer to "Configuring the IP Routing Table" on page 75.
VLAN Parameters	
VLANMode	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.

ini File Parameter	Description
VLANNativeVLANID	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.
VLANOamVLANID	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.
VLANControlVLANID	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.
VLANMediaVLANID	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.
VLANNetworkServiceClassPriority	For a description of this parameter, refer to "Configuring the QoS Settings" on page 78.
VLANPremiumServiceClassMediaPriority	For a description of this parameter, refer to "Configuring the QoS Settings" on page 78.
VLANPremiumServiceClassControlPriority	For a description of this parameter, refer to "Configuring the QoS Settings" on page 78.
VlanGoldServiceClassPriority	For a description of this parameter, refer to "Configuring the QoS Settings" on page 78.
VLANBronzeServiceClassPriority	For a description of this parameter, refer to "Configuring the QoS Settings" on page 78.
EnableDNSasOAM	This parameter applies to both Multiple IPs and VLAN mechanisms. Multiple IPs: Determines the network type for DNS services. VLAN: Determines the traffic type for DNS services. ▪ **[1]** = OAMP (default) ▪ **[0]** = Control.
EnableNTPasOAM	This parameter applies to both Multiple IPs and VLAN mechanisms. Multiple IPs: Determines the network type for NTP services. VLAN: Determines the traffic type for NTP services. ▪ **[1]** = OAMP (default) ▪ **[0]** = Control.
VLANSendNonTaggedOnNative	Specify whether to send non-tagged packets on the native VLAN. ▪ **[0]** = Sends priority tag packets (default). ▪ **[1]** = Sends regular packets (with no VLAN tag).
Multiple IPs Parameters	
EnableMultipleIPs	For a description of this parameter, refer to "Configuring the IP Settings" on page 64. **Note:** This parameter is not applicable when configuring multiple interfaces using the *ini* file table parameter InterfaceTable.
LocalMediaIPAddress	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.
LocalMediaSubnetMask	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.
LocalMediaDefaultGW	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.
LocalControlIPAddress	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.

ini File Parameter	Description
LocalControlSubnetMask	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.
LocalControlDefaultGW	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.
LocalOAMIPAddress	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.
LocalOAMSubnetMask	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.
LocalOAMDefaultGW	For a description of this parameter, refer to "Configuring the IP Settings" on page 64.
Multiple Interface Table	
InterfaceTable	This *ini* file table parameter configures the Multiple Interface table for configuring logical IP addresses. The format of this parameter is as follows: [InterfaceTable] FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes, InterfaceTable_IPv6InterfaceMode, InterfaceTable_IPAddress, InterfaceTable_PrefixLength, InterfaceTable_Gateway, InterfaceTable_VlanID, InterfaceTable_InterfaceName; InterfaceTable 0 = 6, 0, 192.168.85.14, 16, 192.168.0.1, 1, myAll; [\InterfaceTable] For example: [InterfaceTable] FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes, InterfaceTable_IPv6InterfaceMode, InterfaceTable_IPAddress, InterfaceTable_PrefixLength, InterfaceTable_Gateway, InterfaceTable_VlanID, InterfaceTable_InterfaceName; InterfaceTable 0 = 0, 0, 192.168.85.14, 16, 0.0.0.0, 1, ManagementIF; InterfaceTable 1 = 2, 0, 200.200.85.14, 24, 0.0.0.0, 200, myControlIF; InterfaceTable 2 = 1, 0, 211.211.85.14, 24, 211.211.85.1, 211, myMediaIF; [\InterfaceTable] The above example, configures three network interfaces (OAMP, Control, and Media applications). **Notes:** ▪ To configure the Multiple Interface table using the Web interface, refer to "Configuring the Multiple Interface Table" on page 68. ▪ For a description of configuring *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
Differential Services. For detailed information on IP QoS via Differentiated Services, refer to "IP QoS via Differentiated Services (DiffServ)" on page 450.	
NetworkServiceClassDiffServ	For a description of this parameter, refer to "Configuring the QoS Settings" on page 78.
PremiumServiceClassMediaDiffServ	For a description of this parameter, refer to "Configuring the QoS Settings" on page 78.
PremiumServiceClassControlDiffServ	For a description of this parameter, refer to "Configuring the QoS Settings" on page 78.

ini File Parameter	Description
GoldServiceClassDiffServ	For a description of this parameter, refer to "Configuring the QoS Settings" on page 78.
BronzeServiceClassDiffServ	For a description of this parameter, refer to "Configuring the QoS Settings" on page 78.
NFS Table Parameter (NFSServers)	
NFSServers	This *ini* file table parameter defines Network File Systems (NFS) so that the device can access a remote server's shared files and directories for loading *cmp*, *ini*, and auxiliary files (using the Automatic Update mechanism). Alternatively, you can configure NFS using the Web interface (refer to "Configuring the NFS Settings" on page 74). The format of this *ini* file table parameter is as follows: [NFSServers] FORMAT NFSServers_**Index** = NFSServers_**HostOrIP**, NFSServers_**RootPath**, NFSServers_**NfsVersion**, NFSServers_**AuthType**, NFSServers_**UID**, NFSServers_**GID**, NFSServers_**VlanType**; [\NFSServers] Where, ▪ Index = Row index (0-4) of the remote file system. ▪ HostOrIP = Domain name or IP address of the NFS server. ▪ RootPath = Path to the root of the remote file system in the format: /[path]. For example, '/audio'. ▪ NfsVersion = NFS version 2 or 3. ▪ AuthType = Authentication method: 0 (NULL) or 1 (UNIX). ▪ UID = User ID (0 to 65537) for UNIX authentication. ▪ GID = Group ID (0 to 65537) for UNIX authentication. ▪ VlanType= VLAN type for accessing remote file system: 0 (OAMP) or 1 (Media). For example: [NFSServers] FORMAT NFSServers_Index = NFSServers_HostOrIP, NFSServers_RootPath, NFSServers_NfsVersion, NFSServers_AuthType, NFSServers_UID, NFSServers_GID, NFSServers_VlanType; NFSServers 1 = 101.1.13, /audio1, 3, 1, 0, 1, 1; [\NFSServers] **Notes:** ▪ You can configure up to five NFS file systems. ▪ The combination of Host / IP and Root Path must be unique for each index in the table. For example, the table must include only one index entry with a Host / IP of '192.168.1.1' and Root Path of '/audio'. ▪ This parameter is applicable only if VLANs are enabled or if Multiple IPs is configured. ▪ For a description of configuring *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.

4.4.2 System Parameters

Table 4-2: System Parameters

ini File Parameter	Description
GroundKeyDetection	Enables analog ground-key detection (FXS and FXO modules implement ground-start signaling) per device. When disabled, the device uses loop-start signaling. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable (enables ground start) **Notes:** ▪ This parameter is applicable only to Mediant 1000. ▪ For ground-start signaling, ensure that the FXO G module is installed (and not the regular FXO module) in the Mediant 1000 chassis. ▪ For FXO ground-start signaling, ensure that the parameters EnableCurrentDisconnect is set to 1 and FXOBetweenRingTime is set to 300.
EnableDiagnostics	Checks the correct functionality of the different hardware components on the device. On completion of the check, if the test fails, the device sends information on the test results of each hardware component to the Syslog server. ▪ **[0]** = Rapid and Enhanced self-test mode (default). ▪ **[1]** = Detailed self-test mode (full test of DSPs, PCM, Switch, LAN, PHY and Flash). ▪ **[2]** = A quicker version of the Detailed self-test mode (full test of DSPs, PCM, Switch, LAN, PHY, but partial test of Flash). For detailed information, refer to the *Product Reference Manual*.
WatchDogStatus	▪ **[0]** = Disable device's watch dog. ▪ **[1]** = Enable device's watch dog (default). **Note:** This parameter is applicable only to Mediant 1000.
LifeLineType	Defines the Lifeline phone type. The Lifeline phone is available on port 1 of each analog module. The Lifeline is activated upon one of the following options: ▪ **[0]** = Power down (default). ▪ **[1]** = Power down or when link is down (physical disconnect). ▪ **[2]** = Power down or when link is down or on network failure (logical link disconnect). **Notes:** ▪ This parameter is applicable only to Mediant 1000. ▪ To enable Lifeline switching on network failure, LAN watch dog must be activated (EnableLANWatchDog = 1). ▪ This parameter is only applicable to FXS interface.
GWAppDelayTime	For a description of this parameter, refer to "Advanced Parameters" on page 166.

ini File Parameter	Description
ActivityListToLog	The Activity Log mechanism enables the device to send log messages (to a Syslog server) that report certain types of Web actions according to a pre-defined filter. The following filters are available: ▪ **[PVC]** (Parameters Value Change) = Changes made on-the-fly to parameters. ▪ **[AFL]** (Auxiliary Files Loading) = Loading of auxiliary files (e.g., via 'Certificate' screen). ▪ **[DR]** (Device Reset) = Reset of device via the 'Maintenance Actions' screen. ▪ **[FB]** (Flash Memory Burning) = Burning of files / parameters to flash (in 'Maintenance Actions' screen). ▪ **[SWU]** (Device Software Update) = cmp loading via the Software Upgrade Wizard. ▪ **[ARD]** (Access to Restricted Domains) = Access to Restricted Domains. The following screens are restricted: (1) ini parameters (AdminPage) (2) 'General Security Settings' (3) 'Configuration File' (4) 'IPSec/IKE' tables (5) 'Software Upgrade Key' (6) 'Internal Firewall' (7) 'Web Access List' (8) 'Web User Accounts' ▪ **[NAA]** (Non Authorized Access) = Attempt to access the Web interface with a false / empty user name or password. ▪ **[SPC]** (Sensitive Parameters Value Change) = Changes made to sensitive parameters: (1) IP Address (2) Subnet Mask (3) Default Gateway IP Address (4) ActivityListToLog For example: ActivityListToLog = 'pvc', 'afl', 'dr', 'fb', 'swu', 'ard', 'naa', 'spc'
ECHybridLoss	Sets the four wire to two wire worst-case Hybrid loss, the ratio between the signal level sent to the hybrid and the echo level returning from the hybrid. ▪ **[0]** = 6 dB (default) ▪ **[1]** = N/A ▪ **[2]** = 0 dB ▪ **[3]** = 3 dB
GwDebugLevel	For a description of this parameter, refer to "Advanced Parameters" on page 166.
CDRReportLevel	For a description of this parameter, refer to "Advanced Parameters" on page 166.
CDRSyslogServerIP	For a description of this parameter, refer to "Advanced Parameters" on page 166.

ini File Parameter	Description
HeartBeatDestIP	Destination IP address (in dotted format notation) to which the device sends proprietary UDP 'ping' packets. The default IP address is 0.0.0.0.
HeartBeatDestPort	Destination UDP port to which the heartbeat packets are sent. The range is 0 to 64000. The default is 0.
HeartBeatIntervalmsec	Delay (in msec) between consecutive heartbeat packets. ▪ **[10]** = 100000. ▪ **[-1]** = disabled (default).
EnableRAI	▪ **[0]** = Disable RAI (Resource Available Indication) service (default). ▪ **[1]** = Enable RAI service. If RAI is enabled, an SNMP 'acBoardCallResourcesAlarm' Alarm Trap is sent if device's busy endpoints exceed a predefined (configurable) threshold.
RAIHighThreshold	High threshold percentage of total calls that are active (busy endpoints). When the percentage of the device's busy endpoints exceeds this High Threshold, the device sends the SNMP acBoardCallResourcesAlarm Alarm Trap with a 'major' Alarm Status. The range is 0 to 100. The default value is 90. **Note:** The percentage of busy endpoints is calculated by dividing the number of busy endpoints by the total number of “enabled” endpoints (trunks are physically connected and synchronized with no alarms and endpoints are defined in the Trunk Group table).
RAILowThreshold	Low threshold percentage of total calls that are active (busy endpoints). When the percentage of the device's busy endpoints falls below this Low Threshold, the device sends an SNMP acBoardCallResourcesAlarm Alarm Trap with a 'cleared' Alarm Status. The range is 0 to 100%. The default value is 90%.
RAILoopTime	Time interval (in seconds) that the device periodically checks call resource availability. The valid range is 1 to 200. The default is 10.
Disconnect Supervision Parameters	
TelConnectCode	For a description of this parameter, refer to "Advanced Parameters" on page 166.
DisconnectOnBrokenConnection	For a description of this parameter, refer to "Advanced Parameters" on page 166.
BrokenConnectionEventTimeout	For a description of this parameter, refer to "Advanced Parameters" on page 166.
EnableSilenceDisconnect	For a description of this parameter, refer to "Advanced Parameters" on page 166.
FarEndDisconnectSilencePeriod	For a description of this parameter, refer to "Advanced Parameters" on page 166.
FarEndDisconnectSilenceMethod	For a description of this parameter, refer to "Advanced Parameters" on page 166.

ini File Parameter	Description
FarEndDisconnectSilenceThreshold	Threshold of the packet count (in percentages) below which is considered silence by the device. The valid range is 1 to 100%. The default is 8%. **Note:** Applicable only if silence is detected according to packet count (FarEndDisconnectSilenceMethod = 1).
Automatic Update Parameters	
CmpFileURL	Specifies the name of the *cmp* file and the location of the server (IP address or FQDN) from which the device loads a new *cmp* file and updates itself. The *cmp* file can be loaded using HTTP, HTTPS, FTP, FTPS, or NFS. For example: http://192.168.0.1/filename **Notes:** ▪ When this parameter is set in the *ini* file, the device always loads the *cmp* file after it is reset. ▪ The *cmp* file is validated before it's burned to flash. The checksum of the *cmp* file is also compared to the previously-burnt checksum to avoid unnecessary resets. ▪ The maximum length of the URL address is 255 characters.
IniFileURL	Specifies the name of the *ini* file and the location of the server (IP address or FQDN) from which the device loads the *ini* file. The *ini* file can be loaded using: HTTP, HTTPS, FTP, FTPS or NFS. For example: http://192.168.0.1/filename http://192.8.77.13/config<MAC> https://<username>:<password>@<IP address>/<file name> **Notes:** ▪ When using HTTP or HTTPS, the date and time of the *ini* file are validated. Only more recently-dated *ini* files are loaded. ▪ The optional string '<MAC>' is replaced with the device's MAC address. Therefore, the device requests an *ini* file name that contains its MAC address. This option enables loading different configurations for specific devices. ▪ The maximum length of the URL address is 99 characters.
PrtFileURL	Specifies the name of the Prerecorded Tones file and the location of the server (IP address or FQDN) from which it is loaded. For example: http://server_name/file, https://server_name/file. **Note:** The maximum length of the URL address is 99 characters.
CptFileURL	Specifies the name of the CPT file and the location of the server (IP address or FQDN) from which it is loaded. For example: http://server_name/file, https://server_name/file. **Note:** The maximum length of the URL address is 99 characters.
FXSCoeffFileURL	Specifies the name of the FXS coefficients file and the location of the server (IP address or FQDN) from where it is loaded. For example: http://server_name/file, https://server_name/file. The maximum length of the URL address is 99 characters. **Note:** This parameter is applicable only to Mediant 1000.

ini File Parameter	Description
FXOCoeffFileURL	Specifies the name of the FXO coefficients file and the location of the server (IP address or FQDN) from which it is loaded. For example: http://server_name/file, https://server_name/file. The maximum length of the URL address is 99 characters. **Note:** This parameter is applicable only to Mediant 1000.
CasFileURL	Specifies the name of the CAS file and the location of the server (IP address or FQDN) from which it is loaded. For example: http://server_name/file, https://server_name/file. **Note:** The maximum length of the URL address is 99 characters.
TLSRootFileUrl	Specifies the name of the TLS trusted root certificate file and the location URL from where it's downloaded.
TLSCertFileUrl	Specifies the name of the TLS certificate file and the location URL from where it's downloaded.
UserInfoFileURL	Specifies the name of the User Information file and the location of the server (IP address or FQDN) from which it is loaded. For example: http://server_name/file, https://server_name/file. **Note:** The maximum length of the URL address is 99 characters.
AutoUpdateCmpFile	Enables / disables the Automatic Update mechanism for the cmp file. ▪ **[0]** = The Automatic Update mechanism doesn't apply to the cmp file (default). ▪ **[1]** = The Automatic Update mechanism includes the cmp file.
AutoUpdateFrequency	Determines the number of minutes the device waits between automatic updates. The default value is 0 (the update at fixed intervals mechanism is disabled).
AutoUpdatePredefinedTime	Schedules an automatic update to a predefined time of the day. The range is 'HH:MM' (24-hour format). For example: 20:18 **Note:** The actual update time is randomized by five minutes to reduce the load on the Web servers.
ResetNow	Invokes an immediate restart of the device. This option can be used to activate offline (i.e., not on-the-fly) parameters that are loaded via IniFileUrl. ▪ **[0]** = The immediate restart mechanism is disabled (default). ▪ **[1]** = The device immediately restarts after an *ini* file with this parameter set to 1 is loaded.
BootP and TFTP Parameters	
The BootP parameters are special 'Hidden' parameters. Once defined and saved in the flash memory, they are used even if they don't appear in the *ini* file.	
BootPRetries	**Note:** This parameter only takes effect from the next reset of the device. This parameter is used to:

ini File Parameter	Description	
	Set the number of BootP requests the device sends during start-up. The device stops sending BootP requests when either BootP reply is received or number of retries is reached. ▪ **[1]** = 1 BootP retry, 1 sec. ▪ **[2]** = 2 BootP retries, 3 sec. ▪ **[3]** = 3 BootP retries, 6 sec. (default). ▪ **[4]** = 10 BootP retries, 30 sec. ▪ **[5]** = 20 BootP retries, 60 sec. ▪ **[6]** = 40 BootP retries, 120 sec. ▪ **[7]** = 100 BootP retries, 300 sec. ▪ **[15]** = BootP retries indefinitely.	Set the number of DHCP packets the device sends. After all packets were sent, if there's still no reply, the device loads from flash. ▪ **[1]** = 4 DHCP packets ▪ **[2]** = 5 DHCP packets ▪ **[3]** = 6 DHCP packets (default) ▪ **[4]** = 7 DHCP packets ▪ **[5]** = 8 DHCP packets ▪ **[6]** = 9 DHCP packets ▪ **[7]** = 10 DHCP packets ▪ **[15]** = 18 DHCP packets
BootPSelectiveEnable	Enables the Selective BootP mechanism. ▪ **[1]** = Enabled. ▪ **[0]** = Disabled (default). The Selective BootP mechanism (available from Boot version 1.92) enables the device's integral BootP client to filter unsolicited BootP/DHCP replies (accepts only BootP replies that contain the text 'AUDC' in the vendor specific information field). This option is useful in environments where enterprise BootP/DHCP servers provide undesired responses to the device's BootP requests. **Note:** When working with DHCP (DHCPEnable = 1) the selective BootP feature must be disabled.	
BootPDelay	The interval between the device's startup and the first BootP/DHCP request that is issued by the device. ▪ **[1]** = 1 second (default). ▪ **[2]** = 3 second. ▪ **[3]** = 6 second. ▪ **[4]** = 30 second. ▪ **[5]** = 60 second. **Note:** This parameter only takes effect from the next reset of the device.	
ExtBootPReqEnable	▪ **[0]** = Disable (default). ▪ **[1]** = Enable extended information to be sent in BootP request. If enabled, the device uses the vendor specific information field in the BootP request to provide device-related initial startup information such as blade type, current IP address, software version, etc. For a full list of the vendor specific Information fields, refer to the *Product Reference Manual*. The BootP/TFTP configuration utility displays this information in the 'Client Info' column (refer to the *Product Reference Manual*). **Note:** This option is not available on DHCP servers.	

ini File Parameter	Description
Serial Parameters	
DisableRS232	▪ **[0]** = RS-232 serial port is enabled (default). ▪ **[1]** = RS-232 serial port is disabled. The RS-232 serial port can be used to change the networking parameters and view error / notification messages. For information on establishing a serial communications link with the device, refer to the device's Fast Track.
SerialBaudRate	Determines the value of the RS-232 baud rate. The valid range is any value. It is recommended to use the following standard values: 1200, 2400, 9600 (default), 14400, 19200, 38400, 57600, 115200.
SerialData	Determines the value of the RS-232 data bit. ▪ **[7]** = 7-bit. ▪ **[8]** = 8-bit (default).
SerialParity	Determines the value of the RS-232 polarity. ▪ **[0]** = None (default). ▪ **[1]** = Odd. ▪ **[2]** = Even.
SerialStop	Determines the value of the RS-232 stop bit. ▪ **[1]** = 1-bit (default). ▪ **[2]** = 2-bit.
SerialFlowControl	Determines the value of the RS-232 flow control. ▪ **[0]** = None (default). ▪ **[1]** = Hardware.

4.4.3 Web and Telnet Parameters

Table 4-3: Web and Telnet Parameters

ini File Parameter	Description
WebAccessList_x	Defines up to ten IP addresses that are permitted to access the device's Web interface and Telnet interfaces. Access from an undefined IP address is denied. This security feature is inactive (i.e., the device can be accessed from any IP address) when the table is empty. For example: WebAccessList_0 = 10.13.2.66 WebAccessList_1 = 10.13.77.7 The default value is 0.0.0.0 (i.e., the device can be accessed from any IP address). For defining the Web and Telnet Access list using the Web interface, refer to "Configuring the Web and Telnet Access List" on page 114.
WebRADIUSLogin	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.

ini File Parameter	Description
DisableWebTask	▪ **[0]** = Enable Web management (default). ▪ **[1]** = Disable Web management.
ResetWebPassword	Resets the username and password of the primary and secondary accounts to their defaults. ▪ **[0]** = Password and username retain their values (default). ▪ **[1]** = Password and username are reset (for the default username and password, refer to User Accounts). **Note:** The username and password cannot be reset from the Web interface (i.e., via AdminPage or by loading an *ini* file).
WelcomeMessage	This *ini* file table parameter configures the Welcome message that appears after a Web interface login. The format of this parameter is as follows: [WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text WelcomeMessage 1 = "..." ; WelcomeMessage 2 = "..." ; WelcomeMessage 3 = "..." ; [\WelcomeMessage] For Example: [WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text WelcomeMessage 1 = "**********" ; WelcomeMessage 2 = "********* This is a Welcome message ***" ; WelcomeMessage 3 = "**********" ; [\WelcomeMessage] **Notes:** ▪ Each index represents a line of text in the Welcome message box. Up to 20 indexes can be defined. ▪ If this parameter is not configured, no Welcome message box is displayed. ▪ For a description on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
DisableWebConfig	Determines whether the entire Web interface is in read-only mode. ▪ **[0]** = Enables modifications of parameters (default). ▪ **[1]** = Web interface in read-only mode. When in read-only mode, parameters can't be modified. In addition, the following pages can't be accessed: 'Web User Accounts', 'Certificates', 'Regional Settings', 'Maintenance Actions' and all file-loading pages ('Load Auxiliary Files', 'Software Upgrade Wizard', and 'Configuration File'). **Note:** To return to read/write after you have applied read-only using this parameter (set to 1), you need to reboot your device with an *ini* file that doesn't include this parameter, using the BootP/TFTP Server utility (refer to the *Product Reference Manual*).
HTTPport	HTTP port used for Web management (default is 80).

ini File Parameter	Description
ScenarioFileName	Defines the file name of the Scenario file to be loaded to the device. The file name must have the *dat* extension and can be up to 47 characters. For loading a Scenario using the Web interface, refer to "Loading a Scenario to the Device" on page 46.
Telnet Parameters	
TelnetServerEnable	For a description of this parameter, refer to "Configuring the Application Settings" on page 71.
TelnetServerPort	For a description of this parameter, refer to "Configuring the Application Settings" on page 71.
TelnetServerIdleDisconnect	For a description of this parameter, refer to "Configuring the Application Settings" on page 71.
SSHServerEnable	For a description of this parameter, refer to "Configuring the Application Settings" on page 71.
SSHServerPort	For a description of this parameter, refer to "Configuring the Application Settings" on page 71.
Customizing the Web Appearance Parameters For detailed information on customizing the Web interface interface, refer to "Customizing the Web Interface" on page 48.	
UseProductName	Determines whether the UserProductName text string is displayed instead of the default product name. ▪ **[0]** = Disabled (default). ▪ **[1]** = Enables the display of the user-defined UserProductName text string (in the Web interface interface and in the extracted *ini* file). If enabled, the UserProductName text string is displayed instead of the default product name.
UserProductName	Text string that replaces the default product name that appears in the Web interface (upper right-hand corner) and the extracted *ini* file. The default is 'Mediant 1000/Mediant 600'. The string can be up to 29 characters.
UseWebLogo	▪ **[0]** = Logo image is used (default). ▪ **[1]** = Text string is used instead of a logo image. If enabled, AudioCodes' default logo (or any other logo defined by the LogoFileName parameter) is replaced with a text string defined by the WebLogoText parameter.
WebLogoText	Text string that replaces the logo image. The string can be up to 15 characters.
LogoWidth	Width (in pixels) of the logo image. **Note:** The optimal setting depends on the resolution settings. The default value is 441, which is the width of AudioCodes' displayed logo.
LogoFileName	Name of the image file (of type GIF, JPEG, or JPG) containing the user's logo. File name can be up to 47 characters. The logo file name can be used to replace AudioCodes' default Web logo with a user defined logo.

4.4.4 Security Parameters

Table 4-4: Security Parameters

ini File Parameter	Description
EnableMediaSecurity	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.
MediaSecurityBehaviour	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.
EnableSIPS	For a description of this parameter, refer to "General Parameters" on page 166.
TLSVersion	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.
TLSLocalSIPPort	For a description of this parameter, refer to "General Parameters" on page 166.
TLSReHandshakeInterval	Defines the time interval (in minutes) between TLS Re-Handshakes initiated by the device. The interval range is 0 to 1,500 minutes. The default is 0 (i.e., no TLS Re-Handshake).
PeerHostNameVerificationMode	Determines whether the device verifies the Subject Name of a remote certificate when establishing TLS connections. ▪ **[0]** = Disable (default). ▪ **[1]** = Verify Subject Name only when acting as a server for the TLS connection. ▪ **[2]** = Verify Subject Name when acting as a server or client for the TLS connection. When a remote certificate is received and this parameter is not disabled, the SubjectAltName value is compared with the list of available Proxies. If a match is found for any of the configured Proxies, the TLS connection is established. The comparison is performed if the SubjectAltName is either a DNS name (DNSName) or an IP address. If no match is found and the SubjectAltName is marked as 'critical', the TLS connection is not established. If the SubjectAltName is not marked as 'critical' and there is no match, the CN value of the SubjectName field is compared with the parameter TLSRemoteSubjectName. If a match is found, the connection is established. Otherwise, the connection is terminated.
VerifyServerCertificate	Determines whether the device, when acting as client for TLS connections, verifies the Server certificate. The certificate is verified with the Root CA information. ▪ **[0]** = Disable (default). ▪ **[1]** = Enable. **Note:** If Subject Name verification is necessary, the parameter PeerHostNameVerificationMode must be used as well.
OCSPEnable	Enables or disables certificate checking using Online Certificate Status Protocol (OCSP). ▪ **[0]** = Disable (default). ▪ **[1]** = Enable.

ini File Parameter	Description
OCSPServerIP	Defines the IP address of the OCSP server. The default IP address is 0.0.0.0.
OCSPServerPort	Defines the OCSP server's TCP port number. The default port number is 2560.
OCSPDefaultResponse	Determines the default OCSP behavior when the server cannot be contacted. ▪ **[0]** = Rejects peer certificate (default). ▪ **[1]** = Allows peer certificate.
TLSRemoteSubjectName	Defines the Subject Name that is compared with the name defined in the remote side certificate when establishing TLS connections. If the SubjectAltName of the received certificate is not equal to any of the defined Proxies Host names/IP addresses and is not marked as 'critical', the Common Name (CN) of the Subject field is compared with this value. If not equal, the TLS connection is not established. The valid range is a string of up to 49 characters. **Note:** This parameter is applicable only if the parameter PeerHostNameVerificationMode is set to 1 or 2.
EnableSecureStartup	Enables the Secure Startup mode. In this mode, downloading the *.ini file to the device is restricted to a URL provided in initial configuration (see parameter IniFileURL) or using DHCP. ▪ **[0]** Disable (default). ▪ **[1]** Enable = disables TFTP and allows secure protocols such as HTTPS to fetch the device configuration. **Note:** For a detailed explanation on Secure Startup, refer to the *Product Reference Manual*.
SIPSRequireClientCertificate	Determines the device's behavior when acting as a server for TLS connections. ▪ **[0]** = The device does not request the client certificate (default). ▪ **[1]** = The device requires receipt and verification of the client certificate to establish the TLS connection. **Notes:** ▪ The SIPS certificate files can be changed using the parameters HTTPSCertFileName and HTTPSRootFileName. ▪ This parameter cannot be changed on-the-fly and requires a device reset.
IPSec Parameters	
EnableIPSec	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.
IPSecDPDMode	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.
IPSEC_SPD_TABLE	This *ini* file table parameter configures the IPSec SPD table. The format of this parameter is as follows: [IPSEC_SPD_TABLE] Format SPD_INDEX = IPSec**Mode**, IPSec**PolicyRemoteIPAddress**, IPSec**PolicySrcPort**, IPSec**PolicyDStPort**,IPSec**PolicyProtocol**,

ini File Parameter	Description
	IPSec**PolicyLifeInSec**, IPSec**PolicyLifeInKB**, IPSec**PolicyProposalEncryption_X**, IPSec**PolicyProposalAuthentication_X**, IPSec**PolicyKeyExchangeMethodIndex**, IPSec**PolicyLocalIPAddressType**, IPSec**PolicyRemoteTunnelIPAddress**, IPsec**PolicyRemoteSubnetMask**; [\IPSEC_SPD_TABLE] Where, ▪ Mode = IPSec mode of operation: [0] Transport, [1] Tunneling. ▪ PolicyRemoteIPAddress = Destination IP address (or FQDN) to which the IPSec mechanism is applied. ▪ PolicySrcPort = Source port to which the IPSec mechanism is applied. ▪ PolicyDStPort = Destination port to which the IPSec mechanism is applied. ▪ PolicyProtocol = Protocol type to which the IPSec mechanism is applied: [0] Any protocol (default), [17] (UDP), [6] (TCP), any other protocol type defined by IANA (Internet Assigned Numbers Authority). ▪ PolicyLifeInSec = Time (in seconds) that the SA negotiated in the second IKE session (quick mode) is valid. After the time expires, the SA is re-negotiated. ▪ PolicyLifeInKB = Lifetime (in kilobytes) that the SA negotiated in the second IKE session (quick mode) is valid. After this size is reached, the SA is re-negotiated. ▪ PolicyProposalEncryption_X = Encryption type used in the quick mode negotiation for up to four proposals (X depicts the proposal number 0 to 3): [0] No encryption, [1] DES-CBC, [2] Triple DES-CBC, [3] AES-CBC. ▪ PolicyProposalAuthentication_X = Authentication protocol used in the quick mode negotiation for up to four proposals (X depicts the proposal number 0 to 3): [2] HMAC-SHA-1-96, [4] HMAC-MD5-96. ▪ PolicyKeyExchangeMethodIndex = Index for the corresponding IKE entry. ▪ PolicyLocalIPAddressType = Local interface to which the encryption is applied (applicable to multiple IPs and VLANs): [0] OAMP, [1] Control. ▪ PolicyRemoteTunnelIPAddress = IP address of the remote IPSec tunneling device. ▪ PolicyRemoteSubnetMask = subnet mask of the remote IPSec tunneling device. For example: [IPSEC_SPD_TABLE] Format SPD_INDEX = IPSecMode, IPSecPolicyRemoteIPAddress, IpsecPolicySrcPort, IPSecPolicyDStPort,IPSecPolicyProtocol, IPSecPolicyLifeInSec, IPSecPolicyProposalEncryption_0, IPSecPolicyProposalAuthentication_0, IPSecPolicyProposalEncryption_1, IPSecPolicyProposalAuthentication_1, IPSecPolicyKeyExchangeMethodIndex, IPSecPolicyLocalIPAddressType; IPSEC_SPD_TABLE 0 = 0, 10.11.2.21, 0, 0, 17, 900, 1,2, 2,2 ,1, 0; [\IPSEC_SPD_TABLE] In the example above, all packets designated to IP address 10.11.2.21 that originate from the OAMP interface (regardless of their destination and source ports) and whose protocol is UDP are encrypted. The IPSec SPD also defines an SA lifetime of 900 seconds and two security proposals

ini File Parameter	Description
	(DES/SHA1 and 3DES/SHA1). IPsec is performed using the Transport mode. **Notes:** ▪ Each line in the table refers to a different IP destination. ▪ To support more than one Encryption / Authentication proposal, for each proposal specify the relevant parameters in the Format line. ▪ The proposal list must be contiguous. ▪ To configure the IKE table using the Web interface, refer to "Configuring the IPSec Table" on page 126. ▪ For an explanation on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
IKE Parameters	
IPSec_IKEDB_Table	This *ini* file table parameter configures the IKE table. The format of this parameter is as follows: [IPSec_IKEDB_Table] Format IKE_DB_INDEX = IKEPolicy**SharedKey**, IKEPolicy**ProposalEncryption_X**, IKEPolicy**ProposalAuthentication_X**, IKEPolicy**ProposalDHGroup_X**, IKEPolicy**LifeInSec**, IKEPolicy**LifeInKB**, IkePolicy**AuthenticationMethod**; [\IPSEC_IKEDB_TABLE] Where, ▪ SharedKey = Pre-shared key (in textual format). ▪ ProposalEncryption_X = Encryption type for up to four proposals (0 to 3): [1] DES-CBC, [2] Triple DES-CBC, [3] AES-CBC. ▪ ProposalAuthentication_X = Authentication protocol for up to four proposals (0 to 3): [2] HMAC-SHA1-96), [4] HMAC-MD5-96. ▪ ProposalDHGroup_X = Length of key created by the DH protocol for up to four proposals (0 to 3): [0] DH-786-Bit, [1] DH-1024-Bit. ▪ LifeInSec = Time (in seconds) that the SA negotiated in the first IKE session (main mode) is valid. After the time expires, the SA is re-negotiated. ▪ LifeInKB = Lifetime (in kilobytes) that the SA negotiated in the first IKE session (main mode) is valid. After this size is reached, the SA is re-negotiated. ▪ AuthenticationMethod = Authentication method for IKE: [0] Pre-shared Key (default), [1] RSA Signature. For example: [IPSec_IKEDB_Table] Format IKE_DB_INDEX = IKEPolicySharedKey, IKEPolicyProposalEncryption_0, IKEPolicypRoposalAuthentication_0, IKEPolicyProposalDHGroup_0, IKEPolicyProposalEncryption_1, IKEPolicyProposalAuthentication_1, IKEPolicyProposalDHGroup_1, IKEPolicyLifeInSec, IkePolicyAuthenticationMethod; IPSEC_IKEDB_TABLE 0 = 123456789, 1, 2, 0, 2, 2, 1, 28800, 0; [\IPSEC_IKEDB_TABLE] In the example above, a single IKE peer is configured and a pre-shared key authentication is selected. Its pre-shared key is 123456789. Two security proposals are configured: DES/SHA1/786DH and 3DES/SHA1/1024DH

ini File Parameter	Description
	Notes: ▪ Each line in the table refers to a different IKE peer. ▪ To support more than one Encryption / Authentication / DH Group proposal, for each proposal specify the relevant parameters in the Format line. ▪ The proposal list must be contiguous. ▪ To configure the IKE table using the Web interface, refer to "Configuring the IKE Table" on page 130. ▪ For an explanation on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
Secure Hypertext Transport Protocol (HTTPS) Parameters	
HTTPSOnly	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.
HTTPSPort	Determines the local Secured HTTPS port of the device. The valid range is 1 to 65535 (other restrictions may apply within this range). The default port is 443.
HTTPSCipherString	Defines the Cipher string for HTTPS (in OpenSSL cipher list format). Refer to URL http://www.openssl.org/docs/apps/ciphers.html. The range is EXP, RC4. Default is 0.
WebAuthMode	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.
HTTPSRequireClientCertificate	Requires client certificates for HTTPS connection. The client certificate must be preloaded to the device, and its matching private key must be installed on the managing PC. Time and date must be correctly set on the device, for the client certificate to be verified. ▪ **[0]** = Client certificates are not required (default). ▪ **[1]** = Client certificates are required.
HTTPSRootFileName	Defines the name of the HTTPS trusted root certificate file to be loaded via TFTP. The file must be in base64-encoded PEM (Privacy Enhanced Mail) format. The valid range is a 47-character string. **Note:** This parameter is only relevant when the device is loaded via BootP/TFTP. For information on loading this file via the Web interface, refer to the *Product Reference Manual*.
HTTPSPkeyFileName	Defines the name of a private key file (in unencrypted PEM format) to be loaded from the TFTP server.
HTTPSCertFileName	Defines the name of the HTTPS server certificate file to be loaded via TFTP. The file must be in base64-encoded PEM format. The valid range is a 47-character string. **Note:** This parameter is only relevant when the device is loaded using BootP/TFTP. For information on loading this file via the Web interface, refer to the *Product Reference Manual*.
VoiceMenuPassword	For a description of this parameter, refer to Configuring the General Security Settings on page 123. **Note:** This parameter is applicable only to Mediant 1000.

ini File Parameter	Description
Internal Firewall Parameters	
AccessList	This *ini* file table parameter configures the device's access list (firewall), which defines network traffic filtering rules. The format of this parameter is as follows: [ACCESSLIST] FORMAT AccessList_**Index** = AccessList_**Source_IP**, AccessList_**Net_Mask**, AccessList_**Start_Port**, AccessList_**End_Port**, AccessList_**Protocol**, AccessList_**Packet_Size**, AccessList_**Byte_Rate**, AccessList_**Byte_Burst**, AccessList_**Allow_Type**; [\ACCESSLIST] Where, ▪ Source_IP = IP address (or DNS name) of source network, or a specific host. ▪ Net_Mask = IP network mask 255.255.255.255 for a single host or the appropriate value for the source IP addresses. ▪ Start_Port = Start destination UDP/TCP ports (on this device) to which packets are sent. ▪ End_Port = End destination UDP/TCP ports (on this device) to which packets are sent. ▪ Protocol = Protocol type (e.g., UDP, TCP, ICMP, ESP) or the IANA protocol number. ▪ Packet_Size = Maximum allowed packet size. ▪ Byte_Rate = Expected traffic rate (bytes per second). ▪ Byte_Burst = Tolerance of traffic rate limit (number of bytes). ▪ Allow_Type = Action upon match (i.e., 'Allow' or 'Block'). For example: [ACCESSLIST] FORMAT AccessList_Index = AccessList_Source_IP, AccessList_Net_Mask, AccessList_Start_Port, AccessList_End_Port, AccessList_Protocol, AccessList_Packet_Size, AccessList_Byte_Rate, AccessList_Byte_Burst, AccessList_Allow_Type; AccessList 10 = mgmt.customer.com, 255.255.255.255, 0, 80, tcp, 0, 0, 0, allow; AccessList 22 = 10.4.0.0, 255.255.0.0, 4000, 9000, any, 0, 0, 0, block; [\ACCESSLIST] In the example above, Rule #10 allows traffic from the host 'mgmt.customer.com' destined to TCP ports 0 to 80. Rule #22 blocks traffic from the subnet 10.4.xxx.yyy destined to ports 4000 to 9000. **Notes:** ▪ This parameter can include up to 50 indices. ▪ If the end of the table is reached without a match, the packet is accepted. ▪ To configure the firewall using the Web interface, refer to "Structure of ini File Table Parameters" on page 293.
AccessList_MatchCount	For a description of this parameter, refer to "Configuring the Firewall Settings" on page 115.

4.4.5 RADIUS Parameters

For detailed information on the supported RADIUS attributes, refer to "Supported RADIUS Attributes" on page 419.

Table 4-5: RADIUS Parameters

ini File Parameter	Description
EnableRADIUS	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.
AAAIndications	For a description of this parameter, refer to "Configuring RADIUS Accounting Parameters" on page 247.
BehaviorUponRadiusTimeout	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.
MaxRADIUSSessions	Number of concurrent calls that can communicate with the RADIUS server (optional). The valid range is 0 to 240. The default value is 240.
SharedSecret	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.
RADIUSRetransmission	Number of retransmission retries. The valid range is 1 to 10. The default value is 3.
RadiusTO	Determines the time interval (measured in seconds) the device waits for a response before a RADIUS retransmission is issued. The valid range is 1 to 30. The default value is 10.
RADIUSAuthServerIP	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.
RADIUSAuthPort	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.
RADIUSAccServerIP	For a description of this parameter, refer to "Configuring RADIUS Accounting Parameters" on page 247.
RADIUSAccPort	For a description of this parameter, refer to "Configuring RADIUS Accounting Parameters" on page 247.
RadiusAccountingType	For a description of this parameter, refer to "Configuring RADIUS Accounting Parameters" on page 247.
DefaultAccessLevel	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.
RadiusLocalCacheMode	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.
RadiusLocalCacheTimeout	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.
RadiusVSAVendorID	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.
RadiusVSAAccessAttribute	For a description of this parameter, refer to "Configuring the General Security Settings" on page 123.

4.4.6 SNMP Parameters

Table 4-6: SNMP Parameters

ini File Parameter	Description
DisableSNMP	For a description of this parameter, refer to "Configuring the Management Settings" on page 254.
SNMPPort	The device's local UDP port used for SNMP Get/Set commands. The range is 100 to 3999. The default port is 161.
SNMPTrustedMGR_x	Up to five IP addresses of remote trusted SNMP managers from which the SNMP agent accepts and processes get and set requests. **Notes:** ▪ If no values are assigned to these parameters any manager can access the device. ▪ Trusted managers can work with *all* community strings.
ChassisPhysicalAlias	This object is an 'alias' name for the physical entity as specified by a network manager, and provides a non-volatile 'handle' for the physical entity. The valid range is a string of up to 255 characters.
ChassisPhysicalAssetID	This object is a user-assigned asset tracking identifier for the Mediant 1000 chassis as specified by an EMS, and provides non-volatile storage of this information. The valid range is a string of up to 255 characters.
ifAlias	The textual name of the interface. The value is equal to the ifAlias SNMP MIB object. The valid range is a string of up to 64 characters.
KeepAliveTrapPort	The port to which the keep-alive traps are sent. The valid range is 0 - 65534. The default is port 162.
SendKeepAliveTrap	When enabled, this parameter invokes the keep-alive trap and sends it every 9/10 of the time defined in the parameter defining NAT Binding Default Timeout. ▪ **[0]** = Disable ▪ **[1]** = Enable
SNMPSysOid	Defines the base product system OID. Default is eSNMP_AC_PRODUCT_BASE_OID_D.
SNMPTrapEnterpriseOid	Defines a Trap Enterprise OID. Default is eSNMP_AC_ENTERPRISE_OID. The inner shift of the trap in the AcTrap subtree is added to the end of the OID in this parameter.
acUserInputAlarmDescription	Defines the description of the input alarm.
acUserInputAlarmSeverity	Defines the severity of the input alarm.
AlarmHistoryTableMaxSize	Determines the maximum number of rows in the Alarm History table. The parameter can be controlled by the Config Global Entry Limit MIB (located in the Notification Log MIB). The valid range is 50 to 1000. The default value is 500.

ini File Parameter	Description
SNMP Trap Parameters	
SNMPManagerTableIP_x	For a description of this parameter, refer to "Configuring the SNMP Managers Table" on page 257.
SNMPManagerTrapPort_x	For a description of this parameter, refer to "Configuring the SNMP Managers Table" on page 257.
SNMPManagerTrapUser_x	This parameter can be set to the name of any configured SNMPV3 user to associate with this trap destination. This determines the trap format, authentication level, and encryption level. By default, the trap is associated with the SNMP trap community string.
SNMPManagerIsUsed_x	For a description of this parameter, refer to "Configuring the SNMP Managers Table" on page 257.
SNMPManagerTrapSendingEnable_x	For a description of this parameter, refer to "Configuring the SNMP Managers Table" on page 257.
SNMPTrapManagerHostName	For a description of this parameter, refer to "Configuring the Management Settings" on page 254.
SNMP Community String Parameters	
SNMPReadOnlyCommunityString_x	For a description of this parameter, refer to "Configuring the SNMP Community Strings" on page 258.
SNMPReadWriteCommunityString_x	For a description of this parameter, refer to "Configuring the SNMP Community Strings" on page 258.
SNMPTrapCommunityString	For a description of this parameter, refer to "Configuring the SNMP Community Strings" on page 258.
SNMP v3 Users Parameters	
SNMPUsers	This *ini* file table parameter configures SNMP v3 users. The format of this parameter is as follows: [SNMPUsers] FORMAT SNMPUsers_**Index** = SNMPUsers_**Username**, SNMPUsers_**AuthProtocol**, SNMPUsers_**PrivProtocol**, SNMPUsers_**AuthKey**, SNMPUsers_**PrivKey**, SNMPUsers_**Group**; [\SNMPUsers] Where, ▪ Username = Name of the SNMP v3 user. ▪ AuthProtocol = Authentication protocol. ▪ PrivProtocol = Privacy protocol. ▪ AuthKey = Authentication key. ▪ PrivKey = Privacy key. ▪ Group = Group (read/write) with which the SNMP v3 user is associated. For example: [SNMPUsers] FORMAT SNMPUsers_Index = SNMPUsers_Username, SNMPUsers_AuthProtocol, SNMPUsers_PrivProtocol, SNMPUsers_AuthKey, SNMPUsers_PrivKey, SNMPUsers_Group; SNMPUsers 1 = v3admin1, 1, 0, myauthkey, -, 1; [\SNMPUsers] The example above configures user 'v3admin1' with security level

ini File Parameter	Description
	authNoPriv(2), authentication protocol MD5, authentication text password 'myauthkey', and ReadWriteGroup2. **Notes:** ▪ This parameter can include up to 10 indices. ▪ To configure SNMP v3 users through the Web interface, refer to "Configuring SNMP V3 Users" on page 259. ▪ For an explanation on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.

4.4.7 SIP Configuration Parameters

Table 4-7: SIP Parameters

ini File Parameter	Description
SIPTransportType	For a description of this parameter, refer to "SIP General Parameters" on page 134.
TCPLocalSIPPort	For a description of this parameter, refer to "SIP General Parameters" on page 134.
SIPDestinationPort	For a description of this parameter, refer to "SIP General Parameters" on page 134.
EnableTCPConnectionReuse	For a description of this parameter, refer to "SIP General Parameters" on page 134.
SIPTCPTimeout	For a description of this parameter, refer to "SIP General Parameters" on page 134.
LocalSIPPort	For a description of this parameter, refer to "SIP General Parameters" on page 134.
EnableFaxReRouting	For a description of this parameter, refer to "SIP General Parameters" on page 134.
SIPGatewayName	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
IsProxyUsed	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
ProxyName	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
AlwaysSendToProxy	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
SendInviteToProxy	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
PreferRouteTable	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
EnableProxyKeepAlive	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
ProxyKeepAliveTime	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.

ini File Parameter	Description
DNSQueryType	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
ProxyDNSQueryType	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
ProxyIP	This *ini* file table parameter configures the Proxy Set ID table for configuring up to six Proxy Sets, each with up to five Proxy server IP addresses. The format of this parameter is as follows: [ProxyIP] FORMAT ProxyIp_**Index** = ProxyIp_**IpAddress**, ProxyIp_**TransportType**, ProxyIp_**ProxySetId**; [\ProxyIP] Where, ▪ IpAddress = Proxy server's IP address. ▪ TransportType = Not configured (-1) - the global SIPTransportType type is used; UDP (0); TCP (1); TLS (2). ▪ ProxySetId = ID of the Proxy Set. For example: [ProxyIP] FORMAT ProxyIp_Index = ProxyIp_IpAddress, ProxyIp_TransportType, ProxyIp_ProxySetId; ProxyIp 0 = 10.33.37.77, -1, 0; ProxyIp 1 = 10.8.8.10, 0, 2; ProxyIp 2 = 10.8.8.40, -1, 1; ProxyIp 3 = 10.5.6.7, -1, 1; [\ProxyIP] **Notes:** ▪ This parameter can include up to 30 indices (0-29). ▪ For assigning various attributes (such as Proxy Load Balancing) to each Proxy Set ID, refer to the *ini* file parameter ProxySet. ▪ For configuring the Proxy Set ID table using the Web interface, refer to "Proxy Sets Table" on page 153. ▪ For an explanation on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
ProxySet	This *ini* file table parameter configures the Proxy Set table by assigning various attributes per Proxy Set ID. The format of this parameter is as follows: [ProxySet] FORMAT ProxySet_**Index** = ProxySet_**EnableProxyKeepAlive**, ProxySet_**ProxyKeepAliveTime**, ProxySet_**ProxyLoadBalancingMethod**, ProxySet_**IsProxyHotSwap**; [\ProxySet] For example: [ProxySet] FORMAT ProxySet_Index = ProxySet_EnableProxyKeepAlive, ProxySet_ProxyKeepAliveTime, ProxySet_ProxyLoadBalancingMethod, ProxySet_IsProxyHotSwap; ProxySet 0 = 0, 60, 0, 0; ProxySet 1 = 1, 60, 1, 0; [\ProxySet]

ini File Parameter	Description
	Notes: ▪ This table parameter can include up to 6 indices (0-5). ▪ For configuring the Proxy Sets, refer to the *ini* file parameter ProxyIP. ▪ For configuring the Proxy Set ID table using the Web interface, refer to "Proxy Sets Table" on page 153. ▪ For an explanation on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
UseSIPTgrp	For a description of this parameter, refer to "SIP General Parameters" on page 134.
EnableGRUU	For a description of this parameter, refer to "SIP General Parameters" on page 134.
UserAgentDisplayInfo	For a description of this parameter, refer to "SIP General Parameters" on page 134.
SIPSDPSessionOwner	For a description of this parameter, refer to "SIP General Parameters" on page 134.
RetryAfterTime	For a description of this parameter, refer to "SIP General Parameters" on page 134.
EnablePAssociatedURIHeader	For a description of this parameter, refer to "SIP General Parameters" on page 134.
EnableContactRestriction	For a description of this parameter, refer to "SIP General Parameters" on page 134.
ReRegisterOnConnectionFailure	Enables the device to perform SIP Re-Registration upon TCP/TLS connection failure. ▪ **[0]** Disable (default). ▪ **[1]** Enable.
SourceNumberPreference	Determines the SIP header used to determine the Source Number in incoming INVITE messages. ▪ **“”** (empty string) = Use device's internal logic for header preference (default). ▪ **“FROM”** = Use the Source Number received in the From header. The valid range is a string of up to 10 characters. The default is an empty string.
EnableRTCPAttribute	Enables or disables the use of the 'rtcp' attribute in the outgoing SDP. ▪ **[0]** = Disable ▪ **[1]** = Enable (default)

ini File Parameter	Description
OfferUnencryptedSRTCP	Determines whether the device adds the UNENCRYPTED_SRTCP attribute to outgoing SDP messages when SRTP/SRTCP is enabled. ▪ **[0]** Disable (default). ▪ **[1]** Enable. For example: a=crypto:1 AES_CM_128_HMAC_SHA1_80 inline:WxMz3YE0fcbjVJOGgSxweCZxySisl6SWi103t6No UNENCRYPTED_SRTCP **Note:** When enabled, the device adds the UNENCRYPTED_SRTCP attribute to the 'crypto' attribute of the outgoing SDP (SRTCP is still encrypted). To disable the sending of RTCP packets, set the parameter RTCPInterval to 0 (refer to "Configuring the RTP / RTCP Settings" on page 88).
OPTIONSUserPart	Defines the User-Part value of the Request-URI for outgoing SIP OPTIONS requests. If no value is configured, the endpoint number (analog interfaces) or configuration parameter 'Username' value (digital interfaces) is used. A special value is 'empty', indicating that no User-Part in the Request-URI (Host-Part only) is used. The valid range is a 30-character string. The default value is an empty string ('').
UseGatewayNameForOptions	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
IsProxyHotSwap	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
HotSwapRtx	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
ProxyRedundancyMode	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
ProxyLoadBalancingMethod	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
ProxyIPListRefreshTime	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
IsFallbackUsed	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
UserName	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
Password	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
Cnonce	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
SIPChallengeCachingMode	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
MutualAuthenticationMode	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
IsRegisterNeeded	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.

ini File Parameter	Description
RegistrarIP	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
RegistrarTransportType	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
RegistrarName	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
GWRegistrationName	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
AuthenticationMode	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
OOSOnRegistrationFail	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
RegistrationTime	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
RegistrationTimeDivider	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
RegistrationRetryTime	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
RegisterOnInviteFailure	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
RegistrationTimeThreshold	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
ZeroSDPHandling	Determines the device's response to an incoming SDP with an IP address of 0.0.0.0 in the Connection line. ▪ **[0]** Sets the IP address of the outgoing SDP Connection line to 0.0.0.0 (default). ▪ **[1]** Sets the IP address of the outgoing SDP Connection line to the device's own IP address and adds a 'a=sendonly' line to the SDP.
ForkingHandlingMode	Determines how the device reacts to forking of outgoing INVITE messages by the Proxy. ▪ **[0]** = Sequential. The device opens a voice stream toward the first 18x SIP response that includes an SDP, and disregards any 18x response with an SDP received thereafter (default). ▪ **[1]** = Parallel. The device opens a voice stream toward the first 18x SIP response that includes an SDP, and re-opens the stream toward any subsequent 18x responses with an SDP. **Note:** Regardless of the ForkingHandlingMode value, once a 200 OK response is received, the device uses the RTP information and re-opens the voice stream, if necessary.
Account	This *ini* file table parameter configures the Account table for registering and/or authenticating (digest) a Trunk Group (e.g., IP-PBX) to a Serving IP Group (e.g., Internet Telephony Service Provider - ITSP). The format of this parameter is as follows: [Account] FORMAT Account_Index = Account_**ServedTrunkGroup**, Account_**ServedIPGroup**, Account_**ServingIPGroup**, Account_**Username**, Account_**Password**, Account_**HostName**,

ini File Parameter	Description
	Account_**Register**, Account_**ContactUser**; [\Account] Where, ▪ ServedTrunkGroup = Trunk Group ID for which the device performs registration/authentication to a destination IP Group. ▪ ServedIPGroup = Currently not applicable (see note below). ▪ ServingIPGroup = Destination IP (Group) to where the device sends the REGISTER requests (and/or digest authentication username and password) for registering (and/or authenticating) the Trunk Group. ▪ Username = Digest authentication user name. ▪ Password = Digest authentication password. ▪ HostName = Register request URI host name sent by the device to the Serving IP Group. ▪ Register = Enables registration mode (i.e., device sends REGISTER requests to Serving IP Group). ▪ ContactUser = AOR user name. For example: [Account] FORMAT Account_Index = Account_ServedTrunkGroup, Account_ServedIPGroup, Account_ServingIPGroup, Account_Username, Account_Password, Account_HostName, Account_Register, Account_HostName; Account 0 = 1, -1, 1, user, 1234, acl, 1, ITSP1; [\Account] **Notes:** ▪ This table can include up to 10 indices. ▪ The table item Account_ServedIPGroup is currently not applicable and must be left empty (or assigned the value -1). It is used only for IP-to-IP routing applications (supported in the next applicable release). ▪ You can define multiple table indices having the same ServedTrunkGroup with different ServingIPGroups, username, password, HostName, and ContactUser. This provides the capability for registering the same Trunk Group to several ITSP's (i.e., Serving IP Groups). ▪ For configuring the Account table using the Web interface, refer to "Configuring the Account Table" on page 232. ▪ For an explanation on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
IPGroup	This *ini* file table parameter configures the IP Group table. The format of this parameter is as follows: [IPGroup] FORMAT IPGroup_**Index** = IPGroup_**Type**, IPGroup_**Description**, IPGroup_**ProxySetId**, IPGroup_**SIPGroupName**, IPGroup_**ContactUser**, IPGroup_**EnableSurvivability**, IPGroup_**ServingIPGroup**, IPGroup_**SendInviteToProxy**, IPGroup_**AlwaysUseRouteTable**; [\IPGroup]

ini File Parameter	Description
	Where, ▪ Description = Brief description of the IP Group. ▪ ProxySetId = Proxy Set ID associated with the IP Group. ▪ SIPGroupName = Request URI host name used in INVITE and REGISTER messages. ▪ SendInviteToProxy = If enabled (1), the INVITE as a result of REFER/3xx messages are sent to the IP Group, overriding the Refer-To/Contact destination. ▪ AlwaysUseRouteTable = If enabled (1), the device uses the IP address (or domain name) defined in the 'Tel to IP Routing' table (refer to "Tel to IP Routing Table" on page 200) as the Request URI host name in outgoing INVITE messages, instead of the value entered in the SIPGroupName. For example: [IPGroup] FORMAT IPGroup_Index = IPGroup_Type, IPGroup_Description, IPGroup_ProxySetId, IPGroup_SIPGroupName, IPGroup_ContactUser, IPGroup_EnableSurvivability, IPGroup_ServingIPGroup, IPGroup_SendInviteToProxy, IPGroup_AlwaysUseRouteTable; IPGroup 1 = 0, "Verizon gateway", 1, firstIPgroup, , 0, -1, 0, 0; IPGroup 2 = 0, "Avaya server", 2, secondIPgroup, , 0, -1, 0, 0; IPGroup 3 = 0, "IP phones", 1, thirdIPGroup, , 0, -1, 0, 0; [\IPGroup] **Notes:** ▪ This table parameter can include up to 9 indices (1-9). ▪ The parameters IPGroup_Type, IPGroup_ContactUser, IPGroup_EnableSurvivability, and IPGroup_ServingIPGroup are currently not applicable and must be left empty (or -1). These parameters are used only for IP-to-IP call routing applications (supported in the next applicable release). ▪ For configuring the IP Group table using the Web interface, refer to "Configuring the IP Groups" on page 231. ▪ For an explanation on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
NumberOfActiveDialogs	Defines the maximum number of active SIP dialogs that are not call related (i.e., REGISTER and SUBSCRIBE). This parameter is used to control the Registration / Subscription rate. The valid range is 1 to 20. The default value is 20.
PrackMode	For a description of this parameter, refer to "SIP General Parameters" on page 134.
AssertedIdMode	For a description of this parameter, refer to "SIP General Parameters" on page 134.
PAssertedUserName	Defines a 'representative number' (up to 50 characters) that is used as the User Part of the Request-URI in the P-Asserted-Identity header of an outgoing INVITE (for Tel-to-IP calls). The default value is NULL.

ini File Parameter	Description
UseAORInReferToHeader	Defines the source for the SIP URI set in the Refer-To header of outgoing REFER messages. ▪ **[0]** = Use SIP URI from Contact header of the initial call (default). ▪ **[1]** = Use SIP URI from To/From header of the initial call.
UseTelURIForAssertedID	For a description of this parameter, refer to "SIP General Parameters" on page 134.
EnableRPIheader	For a description of this parameter, refer to "SIP General Parameters" on page 134.
IsUserPhone	For a description of this parameter, refer to "SIP General Parameters" on page 134.
IsUserPhoneInFrom	For a description of this parameter, refer to "SIP General Parameters" on page 134.
IsUseToHeaderAsCalledNumber	Determines whether the called number is set in the user part of the To header. ▪ **[0]** = Sets the destination number to the user part of the Request-URI for IP-to-Tel calls, and sets the Contact header to the source number for Tel-to-IP calls (default). ▪ **[1]** = Sets the destination number to the user part of the To header for IP-to-Tel calls, and sets the Contact header to the username parameter for Tel-to-IP calls.
EnableHistoryInfo	For a description of this parameter, refer to "SIP General Parameters" on page 134.
SIPSubject	For a description of this parameter, refer to "SIP General Parameters" on page 134.
MultiPtimeFormat	For a description of this parameter, refer to "SIP General Parameters" on page 134.
EnableReasonHeader	For a description of this parameter, refer to "SIP General Parameters" on page 134.
EnableSemiAttendedTransfer	For a description of this parameter, refer to "SIP General Parameters" on page 134.
EnablePtime	Determines whether the ptime header is included in the SDP. ▪ **[0]** = Remove the ptime header from SDP. ▪ **[1]** = Include the ptime header in SDP (default).
EnableUserInfoUsage	For a description of this parameter, refer to "Advanced Parameters" on page 166.
HandleReasonHeader	Determines whether the device uses the value of the incoming SIP Reason header for Release Reason mapping. ▪ **[0]** Disregard Reason header in incoming SIP messages. ▪ **[1]** Use the Reason header value for Release Reason mapping (default).

ini File Parameter	Description
EnableSilenceSuppInSDP	Determines the device's behavior upon receipt of SIP Re-INVITE messages that include the silencesupp:off attribute. ▪ **[0]** = Disregard the silecesupp attribute (default). ▪ **[1]** = Handle incoming Re-INVITE messages that include the silencesupp:off attribute in the SDP as a request to switch to the Voice-Band-Data (VBD) mode.
EnableRport	Enables / disables the usage of the 'rport' parameter in the Via header. ▪ **[0]** = Enabled. ▪ **[1]** = Disabled (default). The device adds an 'rport' parameter to the Via header of each outgoing SIP message. The first Proxy that receives this message sets the 'rport' value of the response to the actual port from which the request was received. This method is used, for example, to enable the device to identify its port mapping outside a NAT. If the Via doesn't include 'rport' tag, the destination port of the response is taken from the host part of the Via header. If the Via includes 'rport' tag without a port value, the destination port of the response is the source port of the incoming request. If the Via includes 'rport' tag with a port value (rport=1001), the destination port of the response is the port indicated in the 'rport' tag.
VBRCoderHeaderFormat	Defines the format of the RTP header for VBR coders. ▪ **[0]** = Payload only (no header, no TOC, no m-factor) -- similar to RFC 3558 Header Free format (default). ▪ **[1]** = Supports RFC 2658 - 1 byte for interleaving header (always 0), TOC, no m-factor. ▪ **[2]** = Payload including TOC only, allow m-factor. ▪ **[3]** = RFC 3558 Interleave/Bundled format.
TransparentCoderOnDataCall	▪ **[0]** = Only use coders from the coder list (default). ▪ **[1]** = Use transparent coder for data calls (according to RFC 4040). The 'Transparent' coder can be used on data calls. When the device receives a Setup message from the ISDN with 'TransferCapabilities = data', it can initiate a call using the coder 'Transparent' (even if the coder is not included in the coder list). The initiated INVITE includes the following SDP attribute: a=rtpmap:97 CLEARMODE/8000 The default Payload Type is set according to the CoderName table. If the Transparent coder is not set in the Coders table, the default value is set to 56. The Payload Type is negotiated with the remote side, i.e., the selected Payload Type is according to the remote side selection. The receiving device must include the 'Transparent' coder in its coder list.
IsFaxUsed	For a description of this parameter, refer to "SIP General Parameters" on page 134.
T38UseRTPPort	Defines the port (with relation to RTP port) for sending and receiving T.38 packets. ▪ **[0]** = Use the RTP port +2 to send / receive T.38 packets (default). ▪ **[1]** = Use the same port as the RTP port to send / receive T.38 packets.

ini File Parameter	Description
DefaultReleaseCause	For a description of this parameter, refer to "SIP General Parameters" on page 134.
IPAlertTimeout	For a description of this parameter, refer to "SIP General Parameters" on page 134.
SIPPSessionExpires	For a description of this parameter, refer to "SIP General Parameters" on page 134.
SessionExpiresMethod	For a description of this parameter, refer to "SIP General Parameters" on page 134.
MINSE	For a description of this parameter, refer to "SIP General Parameters" on page 134.
SIPMaxRtx	For a description of this parameter, refer to "SIP General Parameters" on page 134.
SipT1Rtx	For a description of this parameter, refer to "SIP General Parameters" on page 134.
SipT2Rtx	For a description of this parameter, refer to "SIP General Parameters" on page 134.
EnableEarlyMedia	For a description of this parameter, refer to "SIP General Parameters" on page 134.
IgnoreAlertAfterEarlyMedia	Determines the device's interworking of ALERT messages from PRI to SIP. ▪ **[0]** = Disabled (default). ▪ **[1]** = Enabled. When enabled, if the device already sent a 183 response with an SDP included and an ALERT message is received from the Tel side (with or without Progress Indicator), the device does not send an additional 18x response and the voice channel remains open. When disabled, the device sends additional 18x responses as a result of receiving an ALERT message whether or not a 18x response was already sent.
EnableTransfer	For a description of this parameter, refer to "Supplementary Services" on page 176.
XferPrefix	For a description of this parameter, refer to "Supplementary Services" on page 176.
EnableHold	For a description of this parameter, refer to "Supplementary Services" on page 176.
HoldFormat	For a description of this parameter, refer to "Supplementary Services" on page 176.
EnableForward	For a description of this parameter, refer to "Supplementary Services" on page 176.
CallWaitingPerPort	This ini file table parameter defines call waiting per port. The format of this parameter is as follows: [CallWaitingPerPort] FORMAT CallWaitingPerPort_Index = CallWaitingPerPort_IsEnabled; CallWaitingPerPort_Port, CallWaitingPerPort_Module; [\CallWaitingPerPort]

ini File Parameter	Description
	Where, ▪ IsEnabled = Enables [1] or disables [0] call waiting. ▪ Port = Port number. ▪ Module = Module number. For example: [CallWaitingPerPort] CallWaitingPerPort 0 = 0,1,1$$; CallWaitingPerPort 1 = 1,2,1$$; [\CallWaitingPerPort] If enabled, when an FXS interface receives a call on a busy endpoint, it responds with a 182 response (and not with a 486 busy). The device plays a call waiting indication signal. When hook-flash is detected, the device switches to the waiting call. The device that initiates the waiting call plays a Call Waiting Ringback tone to the calling party after a 182 response is received. **Notes:** ▪ This parameter is applicable only to Mediant 1000. ▪ If this parameter is not configured (default), use the global parameter EnableCallWaiting (refer to "Supplementary Services" on page 176). ▪ The numbering of channels starts at 0. ▪ The device's Call Progress Tones file must include a 'call waiting Ringback' tone (caller side) and a 'call waiting' tone (called side, FXS interfaces only). ▪ The EnableHold parameter must be enabled on both the calling and the called sides. ▪ To define call waiting using the Web interface, refer to "Call Waiting" on page 440. ▪ For a description on using ini file table parameters, refer to Structure of ini File Table Parameters on page 293.
CHRRTimeout	For a description of this parameter, refer to "Supplementary Services" on page 176.
EnableCallWaiting	For a description of this parameter, refer to "Supplementary Services" on page 176.

ini File Parameter	Description
3WayConferenceMode	Defines the mode of operation when the 3-Way Conference feature is used. ▪ **[0]** = Conference-initiating INVITE (sent by the device), uses the ConferenceID concatenated with a unique identifier as the Request-UR (default). ▪ **[1]** = Conference-initiating INVITE (sent by the device), uses only the ConferenceID as the Reques-URI. If 3wayConferenceMode is set to 0, the Conference-initiating INVITE sent by the device uses the ConferenceID concatenated with a unique identifier as the Request-URI. This same Request-URI is set as the Refer-To header value in the REFER messages that are sent to the two remote parties. If 3wayConferenceMode is set to 1, the Conference-initiating INVITE sent by the device only uses the ConferenceID as the Reques-URI. The media server sets the Contact header of the 200 OK response to the actual unique identifier (Conference URI) to be used by the participants. This Conference URI is included (by the device) in the Refer-To header value in the REFER messages sent by the device to the remote parties. The remote parties join the conference by sending INVITE messages to the media server using this conference URI. **Note:** This parameter is applicable only to Mediant 1000.
Enable3WayConference	For a description of this parameter, refer to "Supplementary Services" on page 176. **Note:** This parameter is applicable only to Mediant 1000.
ConferenceCode	For a description of this parameter, refer to "Supplementary Services" on page 176. **Note:** This parameter is applicable only to Mediant 1000.
ConferenceID	For a description of this parameter, refer to "Supplementary Services" on page 176. **Note:** This parameter is applicable only to Mediant 1000.
BipOnConference	Determines whether a beep is played when a new participant joins a conference and when a participant leaves a conference (in the latter case, a beep of a different pitch is heard). ▪ **[0]** = Beep is disabled. ▪ **[1]** = Beep is enabled (default). **Note:** This parameter is applicable only to Mediant 1000.
Send180ForCallWaiting	Determines the SIP response code for indicating call waiting. ▪ **[0]** = Use 182 Queued response to indicate call waiting (default). ▪ **[1]** = Use 180 Ringing response to indicate call waiting.
HookFlashCode	For a description of this parameter, refer to "Supplementary Services" on page 176.
UseSIPURIForDiversionHeader	Sets the URI format in the SIP Diversion header. ▪ **[0]** = 'tel:' (default) ▪ **[1]** = 'sip:'

ini File Parameter	Description
WarningToneDuration	Defines the duration (in seconds) for which Off-Hook Warning Tone is played to the user. The valid range is -1 to 2,147,483,647. The default is 600. **Notes:** ▪ This parameter is applicable only to Mediant 1000. ▪ A negative value indicates that the tone is played infinitely.
FirstCallWaitingToneID	Determines the index of the first Call Waiting Tone in the CPT file. This feature enables the called party to distinguish between four different call origins (e.g., external vs. internal calls). The device plays the tone received in the 'play tone CallWaitingTone#' parameter of an INFO message + the value of this parameter - 1. The valid range is -1 to 100. The default value is -1 (not used). **Notes:** ▪ This parameter is applicable only to Mediant 1000. ▪ It is assumed that all Call Waiting Tones are defined in sequence in the CPT file. ▪ This feature is relevant only to Broadsoft's application servers (the tone is played using INFO message).
RTPOnlyModeForTrunk_ID	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
RTPOnlyMode	For a description of this parameter, refer to "Advanced Parameters" on page 166.
TimeoutBetween100And18x	Defines the timeout (in msec) between receiving a 100 Trying response and a subsequent 18x response. If a 18x response is not received before this timer expires, the call is disconnected. The valid range is 0 to 32,000. The default value is 0 (i.e., no timeout).
TransparentCoderPresentation	Determines the format of the Transparent coder representation in the SDP. ▪ **[0]** = clearmode (default) ▪ **[1]** = X-CCD
RxDTMFOption	For a description of this parameter, refer to "DTMF & Dialing Parameters" on page 160.

ini File Parameter	Description
TxDTMFOption	This *ini* file table parameter determines a single or several (up to 5) preferred transmit DTMF negotiation methods. The format of this parameter is as follows: [TxDTMFOption] FORMAT TxDTMFOption_Index = TxDTMFOption_**Type**; [\TxDTMFOption] For example: [TxDTMFOption] TxDTMFOption 0 = 1; [\TxDTMFOption] **Notes:** ▪ DTMF negotiation methods are prioritized according to the order of their appearance. ▪ When out-of-band DTMF transfer is used ([1], [2], or [3]), the parameter DTMFTransportType is automatically set to 0 (DTMF digits are erased from the RTP stream). ▪ When RFC 2833 ([4]) is used, the device: 1) Negotiates RFC 2833 Payload Type (PT) using local and remote SDPs. 2) Sends DTMF packets using RFC 2833 PT according to the PT in the received SDP. 3) Expects to receive RFC 2833 packets with the same PT as configured by the parameter RFC2833PayloadType. 4) Uses the same PT for send and receive if the remote party doesn't include the RFC 2833 DTMF PT in its SDP. ▪ When TxDTMFOption is set to [0], the RFC 2833 PT is set according to the parameter RFC2833PayloadType for both transmit and receive. ▪ For defining this parameter using the Web interface, refer to "DTMF & Dialing Parameters" on page 160. ▪ For a description on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
DisableAutoDTMFMute	Enables / disables the automatic muting of DTMF digits when out-of-band DTMF transmission is used. ▪ **[0]** = Automatic mute is used (default). ▪ **[1]** = No automatic mute of in-band DTMF. When DisableAutoDTMFMute = 1, the DTMF transport type is set according to the parameter DTMFTransportType and the DTMF digits aren't muted if out-of-band DTMF mode is selected (TxDTMFOption =1, 2 or 3). This enables the sending of DTMF digits in-band (transparent of RFC 2833) in addition to out-of-band DTMF messages. **Note:** Usually this mode is not recommended.
EnableImmediateTrying	Determines if and when the device sends a 100 Trying response to an incoming INVITE request. ▪ **[0]** = 100 Trying response is sent upon receipt of Proceeding message from the PSTN. ▪ **[1]** = 100 Trying response is sent immediately upon receipt of INVITE request (default).
EnableReasonHeader	For a description of this parameter, refer to "SIP General Parameters" on page 134.

ini File Parameter	Description
3xxBehavior	For a description of this parameter, refer to "SIP General Parameters" on page 134.
EnablePChargingVector	For a description of this parameter, refer to "SIP General Parameters" on page 134.
EnableVMURI	For a description of this parameter, refer to "SIP General Parameters" on page 134.
EmergencyRegretTimeout	For a description of this parameter, refer to Advanced Parameters on page 166. **Note:** This parameter is applicable only to Mediant 1000.
EmergencyNumbers	For a description of this parameter, refer to Advanced Parameters on page 166. **Note:** This parameter is applicable only to Mediant 1000.
MaxActiveCalls	For a description of this parameter, refer to "Advanced Parameters" on page 166.
MaxCallDuration	For a description of this parameter, refer to "Advanced Parameters" on page 166.
EnableBusyOut	For a description of this parameter, refer to "Advanced Parameters" on page 166.
EnableDigitDelivery2IP	For a description of this parameter, refer to "Advanced Parameters" on page 166.
EnableDigitDelivery	For a description of this parameter, refer to "Advanced Parameters" on page 166.

ini File Parameter	Description
Authentication	This ini file table parameter defines a username and password combination for authenticating each device port. The format of this parameter is as follows: [Authentication] FORMAT Authentication_Index = Authentication_UserId, Authentication_UserPassword, Authentication_Port, Authentication_Module; [\Authentication] Where, ▪ UserId = User name ▪ UserPassword = Password ▪ Port = Port number ▪ Module = Module number (0 - 5) For example: [Authentication] Authentication 1 = david,14325,1,0; Authentication 2 = Alex,18552,1,0; Authentication 3 = user1, 1234,1,0; [\Authentication] **Notes:** ▪ This parameter is applicable only to Mediant 1000. ▪ You can omit either the username or password using the sign '$$'. If omitted, the port's phone number is used for authentication. ▪ The indexing of this ini file table parameter starts at 1. ▪ To configure the authentication username and password using the Web interface, refer to Authentication on page 219. ▪ For an explanation on using ini file table parameters, refer to Structure of ini File Table Parameters on page 293.
SITDetectorEnable	Enables or disables Special Information Tone (SIT) detection according to the ITU-T recommendation E.180/Q.35. ▪ **[0]** = Disable (default). ▪ **[1]** = Enable.
SourceIPAddressInput	Determines the IP address which the device uses to decide the source of incoming INVITE messages for IP-to-Tel routing. ▪ **[-1]** = Not configured (default). ▪ **[0]** = Use the IP address received in the Contact header of the incoming INVITE message. ▪ **[1]** = Use the actual IP address (Layer 3) from which the SIP packet was received.
EnableSAS	For a description of this parameter, refer to "Stand-Alone Survivability" on page 188. **Note:** SAS is applicable only to Mediant 1000.
SASLocalSIPUDPPort	For a description of this parameter, refer to "Stand-Alone Survivability" on page 188. **Note:** SAS is applicable only to Mediant 1000.

ini File Parameter	Description
SASDefaultGatewayIP	For a description of this parameter, refer to "Stand-Alone Survivability" on page 188. **Note:** SAS is applicable only to Mediant 1000.
SASRegistrationTime	For a description of this parameter, refer to "Stand-Alone Survivability" on page 188. **Note:** SAS is applicable only to Mediant 1000.
SASShortNumberLength	For a description of this parameter, refer to "Stand-Alone Survivability" on page 188. **Note:** SAS is applicable only to Mediant 1000.
SASLocalSIPTCPPort	For a description of this parameter, refer to "Stand-Alone Survivability" on page 188. **Note:** SAS is applicable only to Mediant 1000.
SASLocalSIPTLSPort	For a description of this parameter, refer to "Stand-Alone Survivability" on page 188. **Note:** SAS is applicable only to Mediant 1000.
Profile Parameters	
CoderName	This *ini* file table parameter defines the device's coder list. This includes up to five groups of coders (consisting of up to five coders per group) that can be associated with IP or Tel profiles ('Coder Group Settings' page in the Web interface -- refer to "Coder Group Settings" on page 212). The first group of coders (indices 0 through 4) is the default coder list and default coder group. The format of this parameter is as follows: [CoderName] FORMAT CoderName_Index = CoderName_**Type**, CoderName_**PacketInterval**, CoderName_**rate**, CoderName_**PayloadType**, CoderName_**Sce**; [\CoderName] Where, ▪ Type = Coder name ▪ PacketInterval = Packetization time ▪ Rate = Packetization rate ▪ PayloadType = Payload type ▪ Sce = Silence suppression mode For example: [CoderName] CoderName 0 = g711Alaw64k, 20,,,0; CoderName 1 = g726, $$, 3, 38, 0; CoderName 2 = g729, 40, 255, 255, 1; [\CoderName] **Notes:** ▪ This parameter can include up to 25 indices (i.e., five coders per five coder groups). ▪ The coder name is case-sensitive. ▪ If silence suppression is not defined for a specific coder, the value defined by the parameter EnableSilenceCompression is used. ▪ The value of several fields is hard-coded according to common

ini File Parameter	Description
	standards (e.g., payload type of G.711 U-law is always 0). Other values can be set dynamically. If no value is specified for a dynamic field, a default value is assigned. If a value is specified for a hard-coded field, the value is ignored. ▪ Only the ptime of the first coder in the defined coder list is declared in INVITE / 200 OK SDP, even if multiple coders are defined. ▪ If the coder G.729 is selected and silence suppression is enabled (for this coder), the device includes the string 'annexb=no' in the SDP of the relevant SIP messages. If silence suppression is set to 'Enable w/o Adaptations', 'annexb=yes' is included. An exception is when the remote device is a Cisco gateway (IsCiscoSCEMode). ▪ For a list of supported coders, refer to "Coders" on page 157. ▪ To configure the 'Coders' table in the Web interface, refer to "Coders" on page 157. ▪ For a description of using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
IPProfile	This *ini* file table parameter configures the IP profiles table. The format of this parameter is as follows: [IPProfile] FORMAT IPProfile_Index = IPProfile_**ProfileName**, IPProfile_**IpPreference**, IPProfile_**CodersGroupID**, IPProfile_**IsFaxUsed***, IPProfile_**JitterBufMinDelay***, IPProfile_**JitterBufOptFactor***, IPProfile_**IPDiffServ***, IPProfile_**SigIPDiffServ***, N/A, IPProfile_**RTPRedundancyDepth**, IPProfile_**RemoteBaseUDPPort**, IPProfile_**CNGmode**, IPProfile_**VxxTransportType**, IPProfile_**NSEMode**, N/A, IPProfile_**PlayRBTone2IP**, IPProfile_**EnableEarlyMedia***, IPProfile_**ProgressIndicator2IP***, IPProfile_**EnableEchoCanceller***, IPProfile_CopyDest2RedirectNumber, IPProfile_**MediaSecurityBehaviour**, IPProfile_**CallLimit**; [\IPProfile] For example: [IPProfile] IPProfile_1 = name1,2,1,0,10,13,15,44,1,1,6000,0,2,0,0,0,1,0,1,0,0,-1; IPProfile_2 = name2,$$,$$,$$

ini File Parameter	Description
	▪ The 'Profile Name' assigned to a Profile index, must enable users to identify it intuitively and easily. ▪ To configure the IP Profile table using the Web interface, refer to "IP Profile Settings" on page 216. ▪ For a description of using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
TelProfile	This *ini* file table parameter configures the Tel Profile Settings table. The format of this parameter is as follows: [TelProfile] FORMAT TelProfile_Index = TelProfile_**ProfileName**, TelProfile_**TelPreference**, TelProfile_**CodersGroupID**, TelProfile_**IsFaxUsed***, TelProfile_**JitterBufMinDelay***, TelProfile_**JitterBufOptFactor***, TelProfile_**IPDiffServ***, TelProfile_**SigIPDiffServ***, TelProfile_**DtmfVolume**, TelProfile_**InputGain**, TelProfile_**VoiceVolume**, TelProfile_EnableReversePolarity, TelProfile_EnableCurrentDisconnect, TelProfile_**EnableDigitDelivery**, TelProfile_**EnableEC**, TelProfile_MWIAnalog, TelProfile_MWIDisplay, TelProfile_**FlashHookPeriod**, TelProfile_**EnableEarlyMedia***, TelProfile_**ProgressIndicator2IP***, TelProfile_**TimeForReorderTone***; [\TelProfile] * = Indicates common parameters used in both IP and Tel profiles. TelPreference = determines the priority of the Profile (1 to 20, where 20 is the highest preference). If both IP and Tel profiles apply to the same call, the coders and other common parameters (indicated with an asterisk) of the preferred Profile are applied to that call. If the preference of the Tel and IP profiles is identical, the Tel Profile parameters are applied. For example: [TelProfile] TelProfile 1 = FaxProfile,1,1,1,40,13,22,33,$$,$$,$$,0,0,0,1,0,0,$$,0,$$; TelProfile 2 = ModemProfile,2,2,0,40,13,$$,$$,$$,$$,$$,$$,$$,$$,0,0, 0,$$, 0,$$; [\TelProfile] **Notes:** ▪ This parameter can appear up to 9 times (i.e., indices 1-9). ▪ Two adjacent dollar signs ('$$') indicates that the parameter's default value is used. ▪ The TelProfile index can be used in the Trunk Group table (TrunkGroup parameter). ▪ The 'Profile Name' assigned to a Profile index must enable users to identify it intuitively and easily. ▪ To configure the Tel Profile table using the Web interface, refer to "Tel Profile Settings" on page 214. ▪ For a description of using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.

4.4.8 Media Server Parameters

Table 4-8: Media Server Parameters

ini File Parameter	Description
MSCMLID	For a description of this parameter, refer to "Configuring the IPmedia Parameters" on page 242.
AmsProfile	Must be set to 1 to use advanced audio.
AASPackagesProfile	Must be set to 3 to use advanced audio.
AmsPrimaryLanguage	Determines the primary language used in the advanced audio package. The default value is “eng”. The languages are according to ISO standard 639-2 language codes.
AmsSecondaryLanguage	Determines the secondary language used in the advanced audio package. The default value is “heb”. The languages are according to ISO standard 639-2 language codes.
AMSAllowUrlAsAlias	Determines whether or not play requests for remote URLs are first verified with local audio segments to determine if any have an alias matching for the URL. If a match is found, the corresponding local audio segment is played. ▪ **[0]** = Always use remote storage (default).\ ▪ **[1]** = Check local storage first. One application for this capability is that of a 'provisioned' cache within the device. For details on provisioning an alias and other audio provisioning capabilities, refer to the Audio Provisioning Server (APS) User's Manual.
VoiceStreamUploadMethod	Defines the HTTP request type for loading the voice stream to the file server. ▪ **[0]** = POST (default). ▪ **[1]** = PUT. **Note:** Applicable only to MSCML recording.
APSEnabled	Indicates whether Voice Prompt index references refer to audio provided by the Audio Provisioning Server (APS), or by the local Voice Prompts file. ▪ **[0]** = APS disabled. Local Voice Prompts file is used. An audio reference in a play request (such as http://localhost/0) indicates that the Voice Prompt at index 0 in the Voice Prompts file is played. ▪ **[1]** = APS enabled (default). An audio reference (such as http://localhost/99) indicates that the audio segment provisioned on the APS with segment ID 99 is played.
NetAnnAnncID	For a description of this parameter, refer to "Configuring the IPmedia Parameters" on page 242.
EnableVoiceStreaming	For a description of this parameter, refer to "Configuring the IPmedia Parameters" on page 242.
VoiceStreamUploadPostURI	Defines the URI used on the POST request to upload voice data from the media server to a Web server.
MediaChannels	For a description of this parameter, refer to "Configuring the IPmedia Parameters" on page 242.
ConferenceID	For a description of this parameter, refer to "Configuring the IPmedia Parameters" on page 242.

ini File Parameter	Description
BipOnConference	For a description of this parameter, refer to "Configuring the IPmedia Parameters" on page 242.
TranscodingID	For a description of this parameter, refer to "Configuring the IPmedia Parameters" on page 242.
ActiveSpeakersNotificationMinInterval	For a description of this parameter, refer to Configuring the IPmedia Settings on page 92.
EnableAGC	For a description of this parameter, refer to "Configuring the IPmedia Settings" on page 92.
AGCGainSlope	For a description of this parameter, refer to "Configuring the IPmedia Settings" on page 92.
AGCRedirection	For a description of this parameter, refer to "Configuring the IPmedia Settings" on page 92.
AGCTargetEnergy	For a description of this parameter, refer to "Configuring the IPmedia Settings" on page 92.
EnableConferenceDTMFClamp	For a description of this parameter, refer to "Configuring the IPmedia Parameters" on page 242.
EnableConferenceDTMFReporting	For a description of this parameter, refer to "Configuring the IPmedia Parameters" on page 242.

4.4.9 Voice Mail Parameters

For detailed information on the Voice Mail application, refer to the CPE Configuration Guide for Voice Mail.

Table 4-9: Voice Mail Parameters

ini File Parameter	Description
VoiceMailInterface	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.
LineTransferMode	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.
WaitForDialTime	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.
MWIOnCode	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.
MWIOffCode	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.
MWISuffixCode	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.
Digit Patterns The following digit pattern parameters apply only to VM applications that use the DTMF communication method. For the available pattern syntaxes, refer to the CPE Configuration Guide for Voice Mail.	
DigitPatternForwardOnBusy	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.

ini File Parameter	Description
DigitPatternForwardOnNoAnswer	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.
DigitPatternForwardOnDND	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.
DigitPatternForwardNoReason	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.
DigitPatternForwardOnBusyExt	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.
DigitPatternForwardOnNoAnswerExt	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.
DigitPatternForwardOnDNDExt	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.
DigitPatternForwardNoReasonExt	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.
DigitPatternInternalCall	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.
DigitPatternExternalCall	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.
TelDisconnectCode	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.
DigitPatternDigitToIgnore	For a description of this parameter, refer to "Configuring the Voice Mail (VM) Parameters" on page 244.

4.4.10 PSTN Parameters

Table 4-10: PSTN Parameters

ini File Parameter	Description
General PSTN Parameters	
PCMLawSelect	For a description of this parameter, refer to "Configuring the TDM Bus Settings" on page 251.
ProtocolType	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
ProtocolType_x	Same as the description for parameter ProtocolType, but for a specific trunk ID (x = 0 - 3).
TraceLevel	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
FramingMethod	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
FramingMethod_x	Same as the description for parameter FramingMethod, but for a specific trunk ID (x = 0 - 3).
TerminationSide	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.

ini File Parameter	Description
TerminationSide_x	Same as the description for parameter TerminationSide, but for a specific trunk ID (x = 0 - 3).
ClockMaster	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
ClockMaster_x	Same as the description for parameter ClockMaster, but for a specific trunk ID (x = 0 - 3).
TDMBusClockSource	For a description of this parameter, refer to "Configuring the TDM Bus Settings" on page 251.
TDMBusPSTNAutoClockEnable	For a description of this parameter, refer to "Configuring the TDM Bus Settings" on page 251.
TDMBusLocalReference	For a description of this parameter, refer to "Configuring the TDM Bus Settings" on page 251.
AutoClockTrunkPriority	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
TDMBusPSTNAutoClockRevertingEnable	For a description of this parameter, refer to "Configuring the TDM Bus Settings" on page 251.
LineCode	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
LineCode_x	Same as the description for parameter LineCode, but for a specific trunk ID (x = 0 - 3).
EnableCallingPartyCategory	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
BChannelNegotiation	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
NFASGroupNumber_x	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
DChConfig_x	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
ISDNNFASInterfaceID_x	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
CASTableIndex_x	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
CASFileName_0 **CASFileName_1** **CASFileName_7**	CAS file name (e.g., 'E_M_WinkTable.dat') that defines the CAS protocol. It is possible to define up to eight different CAS files by repeating this parameter. Each CAS file can be associated with one or more of the device trunks using the parameter CASTableIndex_x.
CASTablesNum	1 to 8. Indicates how many CAS protocol configurations files are loaded.
IdleABCDPattern	For a description of this parameter, refer to "Configuring the TDM Bus Settings" on page 251.
IdlePCMPattern	For a description of this parameter, refer to "Configuring the TDM Bus Settings" on page 251.
LineBuildOut.Loss	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.

ini File Parameter	Description
ISDNRxOverlap_x	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
ISDNRxOverlap	**[0]** = Disabled (default). **[1]** = Enabled. Any number bigger than one = Number of digits to receive. **Notes:** ▪ If enabled, the device receives ISDN called number that is sent in the 'Overlap' mode. ▪ The INVITE to IP is sent only after the number (including 'Sending Complete' Info Element) was fully received (in SETUP and/or subsequent INFO Q.931 messages). For detailed information on ISDN overlap dialing, refer to ISDN Overlap Dialing on page 463.
R2Category	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
HeldTimeout	Determines the time period the device can remain in on-hold. If a Resume (un-hold Re- INVITE) message is received before the timer expires, the call is renewed. If this timer expires, the call is released. ▪ **[-1]** = Indefinitely (default). ▪ **[0 - 2400]** =Time to wait in seconds. Currently, this parameter is applicable only to MFC R2 CAS variants.
CallPriorityMode	Enables Priority Calls handling for analog and digital interfaces.For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
MLPPDefaultNamespace	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
SIPDefaultCallPriority	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
MLPPDiffserv	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235. The valid range is 0 to 63. The default value is 50. **Note:** This parameter is applicable to both analog and digital interfaces.
PreemptionToneDuration	Defines the duration (in seconds) in which the device plays a preemption tone to both the Tel and IP sides if a call was preempted. The valid range is 0 to 60. The default is 3. **Note:** If set to 0, no preemption tone is played.
TrunkLifeLineType	Defines the type of trunk lifeline. Short trunks 1-2, 3-4. ▪ **[0]** = Activate lifeline on power down (default). ▪ **[1]** = Activate lifeline on power down or on detection of LAN disconnect. ▪ **[2]** = Activate lifeline on power down or on detection of LAN disconnect or loss of ping.
TrunkAdministrativeState	Defines the administrative state of a trunk. ▪ **[0]** = Lock the trunk; stops trunk traffic to configure the trunk protocol type. ▪ **[2]** = Unlock the trunk (default); enables trunk traffic.

ini File Parameter	Description
ISDN Flexible Behavior Parameters ISDN protocol is implemented in different Switches / PBXs by different vendors. Several implementations vary a little from the specification. Therefore, to provide a flexible interface that supports these ISDN variants, the ISDN behavior parameters are used.	
ISDNInCallsBehavior	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
ISDNIBehavior	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
ISDNGeneralCCBehavior	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
ISDNOutCallsBehavior	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
ISDNIBehavior_x	Same as the description for parameter ISDNIBehavior, but for a specific trunk ID.
ISDNInCallsBehavior_x	Same as the description for parameter ISDNInCallsBehavior, for a specific trunk ID.
ISDNOutCallsBehavior_x	Same as the description for parameter ISDNOutCallsBehavior, but for a specific trunk ID.
PlayRBTone2Tel	For a description of this parameter, refer to "SIP General Parameters" on page 134.
PlayRBTone2IP	For a description of this parameter, refer to "SIP General Parameters" on page 134.
ProgressIndicator2IP	For a description of this parameter, refer to "Advanced Parameters" on page 166.
TimeForReorderTone	For a description of this parameter, refer to Configuring the FXO Parameters on page 248.
ISDNDisconnectOnBusyTone	Determines whether a call is disconnected upon detection of a busy tone. ▪ **[0]** No = Do not disconnect call upon detection of busy tone. ▪ **[1]** Yes = Disconnect call upon detection of busy tone (default). **Note:** This parameter is applicable only to ISDN protocols. For CAS, use the parameter DisconnectOnBusyTone.
DisconnectOnBusyTone	For a description of this parameter, refer to Configuring the FXO Parameters on page 248
EnableVoiceDetection	For a description of this parameter, refer to Configuring the FXO Parameters on page 248. ▪ **[1]** = The device sends 200 OK (to INVITE) messages when speech/fax/modem is detected. ▪ **[0]** = The device sends 200 OK messages immediately after the device finishes dialing (default). Usually this feature is used only when early media (EnableEarlyMedia) is used to establish voice path before the call is answered. **Notes:** ▪ To activate this feature, set EnableDSPIPMDetectors to 1. ▪ This feature is applicable only when the protocol type is CAS.

ini File Parameter	Description
DigitMapping	For a description of this parameter, refer to "DTMF & Dialing Parameters" on page 160.
TimeBetweenDigits	For a description of this parameter, refer to "DTMF & Dialing Parameters" on page 160.
MaxDigits	For a description of this parameter, refer to "DTMF & Dialing Parameters" on page 160.
TimeForDialTone	For a description of this parameter, refer to "DTMF & Dialing Parameters" on page 160.
RegretTime	For a description of this parameter, refer to "Advanced Parameters" on page 166.

4.4.11 ISDN and CAS Interworking-Related Parameters

Table 4-11: ISDN and CAS Interworking-Related Parameters

ini File Parameter	Description
EnableTDMoverIP	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
EnableISDNTunnelingTel2IP	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
EnableISDNTunnelingIP2Tel	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
ISDNDuplicateQ931BuffMode	Controls the activation / deactivation of delivering raw Q.931 messages. ▪ **[0]** = ISDN messages aren't duplicated (default). ▪ **[128]** = All ISDN messages are duplicated. **Note:** This parameter is not updated on-the-fly and requires a device reset.
EnableQSIGTunneling	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
PlayRBTone2Trunk_ID	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
DigitalOOSBehaviorForTrunk_ID	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
DigitalOOSBehavior	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
RemoveCallingName	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
DefaultCauseMapISDN2IP	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.

ini File Parameter	Description
CauseMapSIP2ISDN	This *ini* file table parameter maps SIP Responses to Q.850 Release Causes. The format of this parameter is as follows: [CauseMapSIP2ISDN] FORMAT CauseMapSIP2ISDN_**Index** = CauseMapSIP2ISDN_**SipResponse**, CauseMapSIP2ISDN_**IsdnReleaseCause**; [\CauseMapSIP2ISDN] Where, ▪ SipResponse = SIP Response ▪ IsdnReleaseCause = Q.850 Release Cause For example: [CauseMapSIP2ISDN] CauseMapSIP2ISDN 0 = 480,50; CauseMapSIP2ISDN 0 = 404,3; [\CauseMapSIP2ISDN] When a SIP response is received (from the IP side), the device searches this mapping table for a match. If the SIP response is found, the Release Cause assigned to it is sent to the PSTN. If no match is found, the default static mapping is used. **Notes:** ▪ This parameter can appear up to 12 times. ▪ For an explanation on *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
CauseMapISDN2SIP	This *ini* file table parameter maps Q.850 Release Causes to SIP Responses. The format of this parameter is as follows: [CauseMapISDN2SIP] FORMAT CauseMapISDN2SIP_**Index** = CauseMapISDN2SIP_**IsdnReleaseCause**, CauseMapISDN2SIP_**SipResponse**; [\CauseMapISDN2SIP] Where, ▪ IsdnReleaseCause = Q.850 Release Cause ▪ SipResponse = SIP Response For example: [CauseMapISDN2SIP] CauseMapISDN2SIP 0 = 50,480; CauseMapISDN2SIP 0 = 6,406; [\CauseMapISDN2SIP] When a Release Cause is received (from the PSTN side), the device searches this mapping table for a match. If the Q.850 Release Cause is found, the SIP response assigned to it is sent to the IP side. If no match is found, the default static mapping is used. **Notes:** ▪ This parameter can appear up to 12 times. ▪ For an explanation on *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.

ini File Parameter	Description
SITQ850Cause	Determines the Q.850 cause value specified in the Reason header that is included in a 4xx response when Special Information Tone (SIT) is detected on an IP-to-Tel call. The valid range is 0 to 127. The default value is 34.
UserToUserHeaderFormat	Determines the format of the User-to-User header. ▪ **[0]** = X-UserToUser (default) ▪ **[1]** = User-to-User
RemoveCLIWhenRestricted	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
ScreeningInd2ISDN	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
ProgressIndicator2ISDN_ID	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
PIForDisconnectMsg_ID	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
ConnectOnProgressInd	Enables the play of announcements from IP to PSTN without the need to answer the Tel-to-IP call. It can be used with PSTN networks that don't support the opening of a TDM channel before an ISDN Connect message is received. ▪ **[0]** = Connect message isn't sent after SIP 183 Session Progress message is received (default). ▪ **[1]** = Connect message is sent after SIP 183 Session Progress message is received.
SIP183Behaviour	For a description of this parameter, refer to "SIP General Parameters" on page 134.
LocalISDNRBSource_ID	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
PSTNAlertTimeout	For a description of this parameter, refer to "Advanced Parameters" on page 166.
TrunkPSTNAlertTimeout_ID	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
ISDNTransferCapability_ID	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
BChannelNegotiationForTrunk_ID	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
SendISDNTransferOnConnect	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.

ini File Parameter	Description
ISDNSubAddressFormat	Determines the format of the 'subaddress' value for ISDN Calling and Called numbers. ▪ **[0]** = ASCII (default). ▪ **[1]** = BCD (Binary Coded Decimal) For IP-to-Tel calls, if the incoming SIP INVITE message includes subaddress values in the 'isub' parameter for the Called Number (in the Request-URI) and/or the Calling Number (in the From header), these values are interworked to the outgoing ISDN SETUP message. If the incoming ISDN SETUP message includes 'subaddress' values for the Called Number and/or the Calling Number, these values are interworked to the outgoing SIP INVITE message.
EnableHold2ISDN	Enables interworking of the Hold/Retrieve supplementary service from SIP to PRI. ▪ **[0]** = Disabled (default) ▪ **[1]** = Enabled **Notes:** ▪ This capability is supported only for QSIG and Euro ISDN variants. ▪ To support interworking of the Hold/Retrieve supplementary service from ISDN to SIP, set the parameter EnableHold to 1.
EnableUUITel2IP	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
EnableUUIIP2Tel	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
ScreeningInd2IP	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
SupportRedirectInFacility	Determines whether the Redirect Number is retrieved from the Facility IE. ▪ **[0]** = Not supported (default). ▪ **[1]** = Supports partial retrieval of Redirect Number (number only) from the Facility IE in ISDN SETUP messages. Applicable to Redirect Number according to ECMA-173 Call Diversion Supplementary Services. **Note:** To enable this feature, ISDNDuplicateQ931BuffMode must be set to 1.
EnableCIC	Determines whether Carrier Identification Code (CIC) is relayed to ISDN. ▪ **[0]** = Do not relay the Carrier Identification Code (CIC) to ISDN (default). ▪ **[1]** = CIC is relayed to the ISDN in Transit Network Selection (TNS) IE. If enabled, the CIC code (received in an INVITE Request-URI) is included in a TNS IE in the ISDN SETUP message. For example: INVITE sip:555666;cic=2345@100.2.3.4 sip/2.0. **Note:** Currently, this feature is supported only in the SIP-to-ISDN direction.

ini File Parameter	Description
EnableAOC	▪ **[0]** = Not used (default). ▪ **[1]** = ISDN Advice of Charge (AOC) messages are interworked to SIP. The device supports receipt of ISDN (Euro ISDN) AOC messages. AOC messages can be received during a call (FACILITY messages) or at the end of a call (DISCONNECT or RELEASE messages). The device converts the AOC messages into SIP INFO (during a call) and BYE (end of a call) messages, using a proprietary AOC SIP header. The device supports both Currency and Pulse AOC messages.
PlayBusyTone2ISDN	For a description of this parameter, refer to "SIP General Parameters" on page 134.
TrunkTransferMode_X	Determines the supported trunk transfer method when a SIP REFER message is received. ▪ **[0]** = Not supported (default). ▪ **[1]** = Supports CAS NFA DMS-100 transfer. When a SIP REFER message is received, the device performs a Blind Transfer by executing a CAS Wink, waits for an acknowledged Wink from the remote side, dials the Refer-to number to the switch, and then releases the call. **Note:** A specific NFA CAS table is required. ▪ **[2]** = Supports ISDN transfer: RLT (DMS-100), TBCT (NI2), ECT (EURO ISDN), and Path Replacement (QSIG). When a SIP REFER message is received, the device performs a transfer by sending FACILITY messages to the PBX with the necessary information on the call's legs that are to be connected. The different ISDN variants use slightly different methods (using FACILITY messages) to perform the transfer. ▪ **[3]** = Supports CAS Normal transfer. When a SIP REFER message is received, the device performs a Blind Transfer by executing a CAS Wink, dialing the Refer-to number to the switch, and then releasing the call. ▪ **[4]** = Supports QSIG Single Step transfer: IP-to-Tel: When a SIP REFER message is received, the device performs a transfer by sending a FACILITY message to the PBX, initiating Single Step transfer. Once a success return result is received, the transfer is completed. Tel-to-IP: When a FACILITY message initiating Single Step transfer is received from the PBX, a REFER message is sent to the IP side. **Notes:** ▪ To use QSIG Path Replacement, the parameter UserToUserHeaderFormat must be set to 1. ▪ To configure Trunk Transfer Mode using the Web interface, refer to "Configuring the Trunk Settings" on page 96.
CASTransportType	For a description of this parameter, refer to "Configuring the Voice Settings" on page 81.

ini File Parameter	Description
CASAddressingDelimiters	Determines if delimiters are added to the dialed address or dialed ANI parameters. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable When this parameter is enabled, delimiters such as '*', '#', and 'ST' are added to the dialed address or dialed ANI parameters. When it is disabled, the address and ANI strings remain without delimiters. **Note:** This parameter is not applicable to BRI interfaces.
CASDelimitersPaddingUsage	Defines the digits string delimiter padding usage per trunk. ▪ **[0]** (default) = default address string padding: '*XXX#' (where XXX is the digit string that begins with '*' and ends with '#', when using padding). ▪ **[1]** = special use of asterisks delimiters: '*XXX*YYY*' (where XXX is the address, YYY is the source phone number, and '*' is the only delimiter padding). **Note:** This parameter is not applicable to BRI interfaces.
CasStateMachineGenerateDigitOnTime	For a description of this parameter, refer to "Configuring the CAS State Machines" on page 109.
CasStateMachineGenerateInterDigitTime	For a description of this parameter, refer to "Configuring the CAS State Machines" on page 109.
CasStateMachineDTMFMaxOnDetectionTime	For a description of this parameter, refer to "Configuring the CAS State Machines" on page 109.
CasStateMachineDTMFMinOnDetectionTime	For a description of this parameter, refer to "Configuring the CAS State Machines" on page 109.
CasStateMachineMaxNumOfIncomingAddressDigits	For a description of this parameter, refer to "Configuring the CAS State Machines" on page 109.
CasStateMachineMaxNumOfIncomingANIDigits	For a description of this parameter, refer to "Configuring the CAS State Machines" on page 109.
CasStateMachineCollectANI	For a description of this parameter, refer to "Configuring the CAS State Machines" on page 109.
CasStateMachineDigitSignalingSystem	For a description of this parameter, refer to "Configuring the CAS State Machines" on page 109.
EnableDSPIPMDetectors	Enables or disables the device's DSP detectors. ▪ **[0]** = Disable (default). ▪ **[1]** = Enable. **Notes:** ▪ The device's Feature Key should contain the 'IPMDetector' DSP option. ▪ When enabled (1), the number of available channels is reduced.
XChannelHeader	For a description of this parameter, refer to Advanced Parameters on page 166.
AddIEinSetup	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.

ini File Parameter	Description
SendIEonTG	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
ISDNDMSTimerT310	Overrides the T310 timer for the DMS-100 ISDN variant. T310 defines the timeout between the reception of a PROCEEDING message and the reception of an ALERTING / CONNECT message. The valid range is 10 to 30. The default value is 10 (seconds). **Note:** Applicable only to Nortel DMS and Nortel MERIDIAN PRI variants (ProtocolType = 14 and 35).
ISDNJapanNTTTimerT3JA	T3_JA timer (in seconds). This parameter overrides the internal PSTN T301 timeout on the Users Side (TE side). If an outgoing call from the device to ISDN is not answered during this timeout, the call is released. The valid range is 10 to 240. The default value is 50. Applicable only to Japan NTT PRI variant (ProtocolType = 16). **Note:** This timer is also affected by the parameter PSTNAlertTimeout.
BriLayer2Mode	For a description of this parameter, refer to "Configuring the Trunk Settings" on page 96.
EnablePatternDetector	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
PDPattern	Defines the patterns that can be detected by the Pattern Detector. The valid range is 0 to 0xFF.
PDThreshold	Defines the number of consecutive patterns to trigger the pattern detection event. The valid range is 0 to 31. The default is 5.
Enable911LocationIdIP2Tel	Enables interworking of Emergency Location Identification from SIP to PRI. ▪ **[0]** = Disabled (default) ▪ **[1]** = Enabled When enabled, the From header received in the SIP INVITE is translated into the following ISDN Information Elements (IE): ▪ Emergency Call Control IE. ▪ Generic Information IE to carry the Location Identification Number information. ▪ Generic Information IE to carry the Calling Geodetic Location information. **Note:** This capability is supported only for the NI-2 ISDN variant.
EarlyAnswerTimeout	Defines the time (in seconds) that the device waits for a CONNECT message from the called party (Tel side) after sending a SETUP message. If the timer expires, the call is answered by sending a 200 OK message (IP side). The valid range is 0 to 600. The default value is 0 (i.e., disabled).

4.4.12 Analog Telephony Parameters

Note: Analog parameters are applicable only to Mediant 1000.

Table 4-12: Analog Telephony Parameters

ini File Parameter	Description
Prefix2ExtLine	Defines a string prefix (e.g., '9') that when dialed from an FXS port causes the device's FXS port to play a secondary dial tone and then restart digit collection. The valid range is a 1-character string. The default is an empty string. **Note:** This parameter is applicable only to Mediant 1000 FXS interfaces.
EnableMOH	Enables the option of using an external audio source, which is connected to the device's AUDIO connector (on the CPU module). When enabled, the device uses the incoming audio from this connector instead of playing the Held Tone defined in the Call Progress Tones (CPT) file. ▪ **[0]** = Disable (default). ▪ **[1]** = Enable. **Note:** EnableHold must be set to 1 to enable this feature.
FirstCallRBTId	Defines the index of the first Ringback Tone in the Call Progress Tones (CPT) file. The valid range is -1 to 1,000. The default value is -1 (play standard Ringback tone).
PrecedenceRingingType	Defines the index of the Precedence Ringing tone in the Call Progress Tones (CPT) file. This tone is used when CallPriorityMode is set to 1 and a Precedence call is received from the IP side. The valid range is -1 to 16. The default value is -1 (plays standard Ringing tone).
FXONumberOfRings	Defines the number of rings before the device's FXO interface answers a call. When set to 0, the FXO seizes the line after one ring. When set to 1, the FXO seizes the line after two rings. The valid range is 0 to 255. The default is 0 seconds. **Note:** If caller ID is enabled, and if the number of rings defined by the parameter RingsBeforeCallerID is greater than the number of rings defined by this parameter, the greater value is used.

ini File Parameter	Description
ChargeCode	This *ini* file table parameter configures metering tones (and their time intervals) that the device's FXS interface generates to the Tel side. The format of this parameter is as follows: [ChargeCode] FORMAT ChargeCode_**Index** = ChargeCode_**EndTime1**, ChargeCode_**PulseInterval1**, ChargeCode_**PulsesOnAnswer1**, ChargeCode_**EndTime2**, ChargeCode_**PulseInterval2**, ChargeCode_**PulsesOnAnswer2**, ChargeCode_**EndTime3**, ChargeCode_**PulseInterval3**, ChargeCode_**PulsesOnAnswer3**, ChargeCode_**EndTime4**, ChargeCode_**PulseInterval4**, ChargeCode_**PulsesOnAnswer4**; [\ChargeCode] Where, ▪ EndTime = Period (1 - 4) end time. ▪ PulseInterval = Period (1 - 4) pulse interval. ▪ PulsesOnAnswer = Period (1 - 4) pulses on answer. For example: [ChargeCode] ChargeCode 1 = 7,30,1,14,20,2,20,15,1,0,60,1; ChargeCode 2 = 5,60,1,14,20,1,0,60,1; ChargeCode 3 = 0,60,1; ChargeCode 0 = 6, 3, 1, 12, 2, 1, 18, 5, 2, 0, 2, 1; [\ChargeCode] **Notes:** ▪ The parameter can appear up to 25 times (i.e., up to 25 different metering rules can be defined). ▪ To configure the Charge Codes table using the Web interface, refer to "Charge Codes Table" on page 183. ▪ For an explanation on configuration using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
TargetOfChannel	This *ini* file table parameter defines telephone numbers that are automatically dialed when a specific port is used. The format of this parameter is as follows: [TargetOfChannel] FORMAT TargetOfChannel_**Index** = TargetOfChannel_**Destination**, TargetOfChannel_**Type**, TargetOfChannel_**Port**, TargetOfChannel_**Module**; [\TargetOfChannel]4 Where, ▪ Destination = Destination phone number. ▪ Type: [1] = Destination phone number is automatically dialed if phone is off-hooked (for FXS interface) or ring signal is applied to port (FXO interface). [0] = automatic dialing is disabled. [2] = enables Hotline - when a phone is off- hooked and no digit is pressed for HotLineToneDuration, the destination phone number is automatically dialed. ▪ Port = Port number. ▪ Module = Module number (0 - 5).

ini File Parameter	Description
	For example: [TargetOfChannel] TargetOfChannel 1 = 1001,1,0,1; (Automatic dialing on Port 1, Module 1) [\TargetOfChannel] **Notes:** ▪ The indexing of this *ini* file table parameter starts at 1. ▪ The numbering of channels starts at 0. ▪ Define this parameter for each device port that implements Automatic Dialing. ▪ To configure the Automatic Dialing Table using the Web interface, refer to "Automatic Dialing" on page 220. ▪ For an explanation on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
CallerDisplayInfo	This *ini* file table parameter enables the device to send Caller ID information to IP when a call is made. The format of this parameter is as follows: [CallerDisplayInfo] FORMAT CallerDisplayInfo_**Index** = CallerDisplayInfo_**DisplayString**, CallerDisplayInfo_**IsCidRestricted**, CallerDisplayInfo_**Port**, CallerDisplayInfo_**Module**; [\CallerDisplayInfo] Where, ▪ DisplayString = Caller ID string. ▪ IsCidRestricted = Restriction - [0] not restricted (default); [1] restricted. ▪ Port = Port number. ▪ Module = Module number (0 - 5). For example: [CallerDisplayInfo] CallerDisplayInfo 1 = Susan C.,0,1,0; (Caller ID on port 1 of first module) [\CallerDisplayInfo] **Notes:** ▪ The indexing of this *ini* file table parameter starts at 1. ▪ The numbering of channels starts with 0. ▪ To configure Caller Display Information using the Web interface, refer to "Caller ID" on page 221. ▪ For a description on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.

ini File Parameter	Description
FwdInfo	This *ini* file table parameter forwards IP-to-Tel calls (using SIP 302 response), based on the device's port to which the call is routed. The format of this parameter is as follows: [FwdInfo] FORMAT FwdInfo_**Index** = FwdInfo_**Type**, FwdInfo_**Destination**, FwdInfo_**NoReplyTime**, FwdInfo_**Port**, FwdInfo_**Module**; [\FwdInfo] Where, ▪ Type = Forward Type (for a list of options, refer to "Call Forward" on page 222). ▪ Destination = Telephone number or URI (number@IP address) to which the call is forwarded. ▪ NoReplyTime = Timeout (in seconds) for No Reply. If you have set the Forward Type for this port to No Answer [3], enter the number of seconds the device waits before forwarding the call to the phone number specified. ▪ Port = Port number. ▪ Module = Module number (0 - 5). For example: [FwdInfo] FwdInfo 1 = 1,1001,$$,2,0; FwdInfo 2 = 1,2003@10.5.1.1,$$,2,0; FwdInfo 3 = 3,2005,30,2,0; [\FwdInfo] **Notes:** ▪ The indexing of this parameter starts at 1. ▪ The device ports starts at 0. ▪ To configure the Call Forward table using the Web interface, refer to "Call Forward" on page 222. ▪ For an explanation on *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
EnableCallerID	This *ini* file table parameter configures Caller ID permissions. The format of this parameter is as follows: [EnableCallerID] FORMAT EnableCallerID_**Index** = EnableCallerID_**IsEnabled,** EnableCallerID_**Port**, EnableCallerID_**Module**; [\EnableCallerID] Where, ▪ IsEnabled = Enables [1] or disables [0] (default) Caller ID. ▪ Port = Port number. ▪ Module = Module number (0-5). For example: [EnableCallerID] EnableCallerID 1 = 1,3,2; EnableCallerID 2 = 0,$$,$$; [\EnableCallerID] **Notes:** ▪ The indexing of this *ini* file table parameter starts at 1.

ini File Parameter	Description
	▪ The numbering of ports starts at 0. ▪ If a port isn't configured, its Caller ID generation / detection are determined according to the global parameter EnableCallerID (described in "Supplementary Services" on page 176). ▪ To configure Call ID Permissions using the Web interface, refer to "Caller ID Permissions" on page 224. ▪ For an explanation on *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
EnableDIDWink	For a description of this parameter, refer to "Advanced Parameters" on page 166.
DelayBeforeDIDWink	For a description of this parameter, refer to "Advanced Parameters" on page 166.
EnableReversalPolarity	For a description of this parameter, refer to "Advanced Parameters" on page 166.
EnableCurrentDisconnect	For a description of this parameter, refer to "Advanced Parameters" on page 166.
CutThrough	For a description of this parameter, refer to "Advanced Parameters" on page 166.
FXSOOSBehavior	For a description of this parameter, refer to "Advanced Parameters" on page 166.
NumberOfWaitingIndications	For a description of this parameter, refer to "Supplementary Services" on page 176.
TimeBetweenWaitingIndications	For a description of this parameter, refer to "Supplementary Services" on page 176.
TimeBeforeWaitingIndication	For a description of this parameter, refer to "Supplementary Services" on page 176.
WaitingBeepDuration	For a description of this parameter, refer to "Supplementary Services" on page 176.
EnableCallerID	For a description of this parameter, refer to "Supplementary Services" on page 176.
CallerIDType	For a description of this parameter, refer to "Supplementary Services" on page 176.
SubscriptionMode	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
EnableMWI	For a description of this parameter, refer to "Supplementary Services" on page 176.
MWIAnalogLamp	For a description of this parameter, refer to "Supplementary Services" on page 176.
MWIDisplay	For a description of this parameter, refer to "Supplementary Services" on page 176.
EnableMWISubscription	For a description of this parameter, refer to "Supplementary Services" on page 176.
MWIServerIP	For a description of this parameter, refer to "Supplementary Services" on page 176.

ini File Parameter	Description
SubscribeRetryTime	For a description of this parameter, refer to "Supplementary Services" on page 176.
MWIServerTransportType	For a description of this parameter, refer to "Supplementary Services" on page 176.
MWIExpirationTime	For a description of this parameter, refer to "Supplementary Services" on page 176.
StutterToneDuration	For a description of this parameter, refer to "Supplementary Services" on page 176.
PayPhoneMeteringMode	For a description of this parameter, refer to "Metering Tones" on page 182.
MeteringType	For a description of this parameter, refer to "Metering Tones" on page 182.
KeyCFUnCond	For a description of this parameter, refer to "Keypad Features" on page 184.
KeyCFNoAnswer	For a description of this parameter, refer to "Keypad Features" on page 184.
KeyCFBusy	For a description of this parameter, refer to "Keypad Features" on page 184.
KeyCFBusyOrNoAnswer	For a description of this parameter, refer to "Keypad Features" on page 184.
KeyCFDoNotDisturb	For a description of this parameter, refer to "Keypad Features" on page 184.
KeyCFDeact	For a description of this parameter, refer to "Keypad Features" on page 184.
KeyCLIR	For a description of this parameter, refer to "Keypad Features" on page 184.
KeyCLIRDeact	For a description of this parameter, refer to "Keypad Features" on page 184.
KeyHotLine	For a description of this parameter, refer to "Keypad Features" on page 184.
KeyHotLineDeact	For a description of this parameter, refer to "Keypad Features" on page 184.
KeyBlindTransfer	For a description of this parameter, refer to "Keypad Features" on page 184.
KeyCallWaitingDeact	For a description of this parameter, refer to "Keypad Features" on page 184.
KeyCallWaiting	For a description of this parameter, refer to "Keypad Features" on page 184.
KeyRejectAnonymousCall	For a description of this parameter, refer to "Keypad Features" on page 184.
KeyRejectAnonymousCallDeact	For a description of this parameter, refer to "Keypad Features" on page 184.

ini File Parameter	Description
BlindTransferDisconnectTimeout	Defines the duration (in milliseconds) for which the device waits for a disconnection from the Tel side after the Blind Transfer Code (KeyBlindTransfer) has been identified. When this timer expires, a SIP REFER message is sent toward the IP side. If this parameter is set to 0, the REFER message is immediately sent. The valid range is 0 to 1,000,000. The default is 0. **Note:** This parameter is applicable only to Mediant 1000.
RejectAnonymousCallPerPort	This *ini* file table parameter determines whether the device rejects incoming anonymous calls on FXS interfaces. The format of this parameter is as follows: [RejectAnonymousCallPerPort] FORMAT RejectAnonymousCallPerPort_Index = RejectAnonymousCallPerPort_Enable, RejectAnonymousCallPerPort_Port, RejectAnonymousCallPerPort_Module; [\RejectAnonymousCallPerPort] Where, ▪ Enable = accept [0] (default) or reject [1] incoming anonymous calls. ▪ Port = Port number. ▪ Module = Module number. For example: [RejectAnonymousCallPerPort] RejectAnonymousCallPerPort 0 = 0,1,$$; RejectAnonymousCallPerPort 1 = 1,2,$$; [\RejectAnonymousCallPerPort] If enabled, when a device's FXS interface receives an anonymous call, it responds with a 433 (Anonymity Disallowed) SIP response. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ This parameter is per FXS port. ▪ This parameter can appear up to 24 times. ▪ The double dollar ($$) symbol represents the default value. ▪ For an explanation on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
IsTwoStageDial	For a description of this parameter, refer to "Configuring the FXO Parameters" on page 248.
IsWaitForDialTone	For a description of this parameter, refer to "Configuring the FXO Parameters" on page 248.
FXOBetweenRingTime	For a description of this parameter, refer to "Configuring the FXO Parameters" on page 248.
RingsBeforeCallerID	For a description of this parameter, refer to "Configuring the FXO Parameters" on page 248.
DisconnectOnDialTone	For a description of this parameter, refer to "Configuring the FXO Parameters" on page 248.
GuardTimeBetweenCalls	For a description of this parameter, refer to "Configuring the FXO Parameters" on page 248.

ini File Parameter	Description
NTTDIDSignallingForm	Determines the type of Direct Inward Dialing (DID) signaling support for NTT (Japan) modem: DTMF- or Frequency Shift Keying (FSK)-based signaling. The devices can be connected to Japan's NTT PBX using 'Modem' DID lines. These DID lines are used to deliver a called number to the PBX. ▪ **[0]** = FSK-based signaling (default) ▪ **[1]** = DTMF-based signaling **Note:** This parameter is applicable only to FXS interfaces.
EnableDID	This *ini* file table parameter enables support for Japan NTT 'Modem' Direct Inward Dialing (DID). FXS interfaces can be connected to Japan's NTT PBX using 'Modem' DID lines. These DID lines are used to deliver a called number to the PBX. The DID signal can be sent alone or combined with an NTT Caller ID signal. The format of this parameter is as follows: [EnableDID] FORMAT EnableDID_**Index** = EnableDID_**IsEnable**, EnableDID_**Port**, EnableDID_**Module**; [\EnableDID] Where, ▪ IsEnable = Enables [1] or disables [0] (default) Japan NTT Modem DID support. ▪ Port = Port number. ▪ Module = Module number. For example: [EnableDID] EnableDID 0 = 1,2,0; [\EnableDID] **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ For an explanation on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
EnableCallerIDTypeTwo	Disables the generation of Caller ID type 2 when the phone is off-hooked. Caller ID type 2, also known as off-hook Caller ID is sent to a currently busy telephone to display the caller ID of the waiting call. ▪ **[0]** = Caller ID type 2 isn't played. ▪ **[1]** = Caller ID type 2 is played (default).
PolarityReversalType	Defines the voltage change slope during polarity reversal or wink. ▪ **[0]** = Soft reverse polarity (default). ▪ **[1]** = Hard reverse polarity. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ Some Caller ID signals use reversal polarity and/or Wink signals. In these cases, it is recommended to set PolarityReversalType to 1 (Hard).

ini File Parameter	Description
CurrentDisconnectDuration	The duration (in msec) of the current disconnect pulse. The range is 200 to 1500. The default is 900. **Notes:** ▪ This parameter is applicable for both FXS and FXO interfaces. ▪ The FXO interface detection range is approximately 200 msec of the parameter's value plus 100. For example, if CurrentDisconnectDuration is 200 msec, then the detection range is 100 to 500 msec.
CurrentDisconnectDefaultThreshold	Determines the line voltage threshold which, when reached, is considered a current disconnect detection. The valid range is 0 to 20 Volts. The default value is 4 Volts. **Note:** Applicable only to FXO interfaces.
TimeToSampleAnalogLineVoltage	Determines the frequency at which the analog line voltage is sampled (after offhook), for detection of the current disconnect threshold. The valid range is 100 to 2500 msec. The default value is 1000 msec. **Note:** Applicable only to FXO interfaces.
AnalogCallerIDTimimgMode	▪ **[0]** = Caller ID is generated between the first two rings (default). ▪ **[1]** = The device attempts to find an optimized timing to generate the Caller ID according to the selected Caller ID type. **Notes:** ▪ Applicable only to FXS interfaces. ▪ When used with distinctive ringing, the Caller ID signal doesn't change the distinctive ringing timing.
BellcoreCallerIDTypeOneSubStandard	Selects the Bellcore Caller ID sub-standard. ▪ **[0]** = Between rings (default). ▪ **[1]** = Not ring related.
ETSICallerIDTypeOneSubStandard	Selects the ETSI FSK Caller ID Type 1 sub-standard (FXS only). ▪ **[0]** = ETSI between rings (default). ▪ **[1]** = ETSI before ring DT_AS. ▪ **[2]** = ETSI before ring RP_AS. ▪ **[3]** = ETSI before ring LR_DT_AS. ▪ **[4]** = ETSI not ring related DT_AS. ▪ **[5]** = ETSI not ring related RP_AS. ▪ **[6]** = ETSI not ring related LR_DT_AS.
ETSIVMWITypeOneStandard	Selects the ETSI Visual Message Waiting Indication (VMWI) Type 1 sub-standard. ▪ **[0]** = ETSI VMWI between rings (default) ▪ **[1]** = ETSI VMWI before ring DT_AS ▪ **[2]** = ETSI VMWI before ring RP_AS ▪ **[3]** = ETSI VMWI before ring LR_DT_AS ▪ **[4]** = ETSI VMWI not ring related DT_AS ▪ **[5]** = ETSI VMWI not ring related RP_AS ▪ **[6]** = ETSI VMWI not ring related LR_DT_AS

ini File Parameter	Description
BellcoreVMWITypeOneStandard	Selects the Bellcore VMWI sub-standard. ▪ **[0]** = Between rings (default). ▪ **[1]** = Not ring related.

4.4.13 Number Manipulation and Routing Parameters

Table 4-13: Number Manipulation and Routing Parameters

ini File Parameter	Description
TrunkGroup	This *ini* file table parameter defines the device's endpoints/Trunks. Alternatively, you can use the Web interface (refer to Configuring the Trunk Group Table on page 226). The format of this parameter is shown below: [TrunkGroup] FORMAT TrunkGroup_**Index** = TrunkGroup_**TrunkGroupNum**, TrunkGroup_**FirstTrunkId**, TrunkGroup_**LastTrunkId**, TrunkGroup_**FirstBChannel**, TrunkGroup_**LastBChannel**, TrunkGroup_**FirstPhoneNumber**, TrunkGroup_**ProfileId**, TrunkGroup_**Module**; [\TrunkGroup] Where, ▪ TrunkGroupNum = Trunk Group ID (1 to 99) ▪ FirstTrunkId = Starting physical trunk number, where 0 is the first trunk ▪ FirstBChannel = Starting B-channel (from 1) ▪ LastBChannel = Ending B-channel (up to 31) ▪ FirstPhoneNumber = Phone number associated with the first channel (optional) ▪ ProfileId = Optional Tel Profile ID (1 to 9) applied to the group of channels ▪ LastTrunkId = Ending physical trunk number ▪ Module = Module number For example: [TrunkGroup] TrunkGroup 1 = 0, 0, 0, 1, 31 ,401, 0; (E1 span) TrunkGroup 1 = 0, 0, 0, 1, 31, $$, 1; TrunkGroup 2 = 1, 2, 2, 1, 24, 3000; (T1 span) TrunkGroup 1 = 0, 0, 3, *, 1000; (4 E1 spans; all B-channels) TrunkGroup 3 = 2, 0, 3, 1, 20, 101, 1; (4 E1 spans; 20 B-channels; module 1) TrunkGroup 1 = 1, 0, 3, 1, 2 ,208, 0 ,1; (4 BRI ports; tel. no 208, module 1) [\TrunkGroup] **Notes:** ▪ This parameter can appear up to four times per module. ▪ To represent all B-channels, use an asterisk ('*'). ▪ For a description on *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.

ini File Parameter	Description
DefaultNumber	For a description of this parameter, refer to "DTMF & Dialing Parameters" on page 160.
ChannelSelectMode	For a description of this parameter, refer to "SIP General Parameters" on page 134.
TrunkGroupSettings	This *ini* file table parameter defines rules for port allocation per Trunk Group. If no rule exists, the global rule defined by the parameter ChannelSelectMode takes effect. The format of this parameter is as follows: [TrunkGroupSettings] FORMAT TrunkGroupSettings_**Index** = TrunkGroupSettings_**TrunkGroupId**, TrunkGroupSettings_**ChannelSelectMode**, TrunkGroupSettings_**RegistrationMode**, TrunkGroupSettings_**GatewayName**,TrunkGroupSettings_**ContactUser**, TrunkGroupSettings_**ServingIPGroup**; [\TrunkGroupSettings] Where, ▪ TrunkGroupId = Trunk Group ID number. ▪ ChannelSelectMode = Channel select mode for the Trunk Group. Available values are identical to those defined by the ChannelSelectMode parameter. ▪ RegistrationMode = Registration mode for the Trunk Group (Per Endpoint [0], Per Gateway [1], or Do Not Register [4]). If not configured [-1], the value of AuthenticationMode is used. ▪ GatewayName = SIPGatewayName used as a hostname in the From header in INVITE and REGISTER messages. If not configured, the parameter SIPGatewayName is used. ▪ ContactUser = User part in contact URI in INVITE, and in From, To and Contact headers in REGISTER. ▪ ServingIPGroup = Serving IP Group ID to where INVITE messages initiated by the Trunk Group endpoints are sent. For example: [TrunkGroupSettings] TrunkGroupSettings 0 = 1, 0, 5, audiocodes, user, 1; TrunkGroupSettings 1 = 2, 1, 0, localname, user1, 2; [\TrunkGroupSettings] **Notes:** ▪ This parameter can include up to 240 indices. ▪ For configuring Trunk Group Settings using the Web interface, refer to "Configuring Trunk Group Settings" on page 228. ▪ For a description on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
AddTrunkGroupAsPrefix	For a description of this parameter, refer to "Routing General Parameters" on page 198.
AddPortAsPrefix	For a description of this parameter, refer to "Routing General Parameters" on page 198.
ReplaceEmptyDstWithPortNumber	For a description of this parameter, refer to Routing General Parameters on page 198.

ini File Parameter	Description
CopyDestOnEmptySource	▪ [0] = Leave Source Number empty (default). ▪ [1] = If the Source Number of a Tel-to-IP call is empty, the Destination Number is copied to the Source Number.
AddNPIandTON2CallingNumber	For a description of this parameter, refer to Routing General Parameters on page 198.
AddNPIandTON2CalledNumber	For a description of this parameter, refer to Routing General Parameters on page 198.
UseSourceNumberAsDisplayName	For a description of this parameter, refer to "Routing General Parameters" on page 198.
UseDisplayNameAsSourceNumber	For a description of this parameter, refer to "SIP General Parameters" on page 134.
AlwaysUseRouteTable	For a description of this parameter, refer to "Proxy & Registration Parameters" on page 146.
Prefix	This *ini* file table parameter configures the 'Tel to IP Routing' table for routing Tel-to-IP calls (i.e., to a destination IP address). The format of this parameter is as follows: [PREFIX] FORMAT PREFIX_**Index** = PREFIX_**DestinationPrefix**, PREFIX_**DestAddress**, PREFIX_**SourcePrefix**, PREFIX_**ProfileId**, PREFIX_**MeteringCode**, PREFIX_**SrcIPGroupID**, PREFIX_**DestHostPrefix**, PREFIX_**DestIPGroupID**, PREFIX_**SrcHostPrefix**, PREFIX_**TransportType**, PREFIX_**SrcTrunkGroupID**; [\PREFIX] Where, ▪ DestinationPrefix = Destination phone prefix. ▪ DestAddress = Destination IP address. ▪ SourcePrefix =Source phone prefix. ▪ ProfileID = Profile ID number. ▪ MeteringCode = Charge code. ▪ DestHostPrefix = N/A. ▪ DestIPGroupID = IP Group (1-9) to where you want to route the Tel-to-IP call. ▪ SrcHostPrefix = N/A. ▪ TransportType = Destination transport type: Not configured (-1), the global SIPTransportType type is used; UDP (0); TCP (1); TLS (2). ▪ SrcTrunkGroupID = N/A. For example: [PREFIX] FORMAT PREFIX_Index = PREFIX_DestinationPrefix, PREFIX_DestAddress, PREFIX_SourcePrefix, PREFIX_ProfileId, PREFIX_MeteringCode, PREFIX_SrcIPGroupID, PREFIX_DestHostPrefix, PREFIX_DestIPGroupID, PREFIX_SrcHostPrefix, PREFIX_TransportType, PREFIX_SrcTrunkGroupID; PREFIX 0 = *, quest, *, 0, 255, -1, , 1, , -1, -1; PREFIX 1 = 20, 10.33.37.77, *, 0, 255, -1, , 2, , 0, -1; PREFIX 2 = 30, 10.33.37.79, *, 1, 255, -1, , -1, , 2, -1; [\PREFIX]

ini File Parameter	Description
	Notes: ▪ This parameter can include up to 50 indices. ▪ The parameters SrcIPGroupID, DestHostPrefix, and SrcHostPrefix are currently not applicable and must be left empty (or -1). (They are used only for IP-to-IP routing, supported in the next applicable release). ▪ The phone prefix for destination (DestinationPrefix) and source (SourcePrefix) addresses can be a single number or a range of numbers. ▪ Parameters can be skipped using two dollar ($$) symbols, for example: Prefix = $$,10.2.10.2,202,1. ▪ The destination IP address (DestAddress) can be either in dotted-decimal notation or FQDN. If an FQDN is used, DNS resolution is performed according to DNSQueryType. ▪ The IP address can include wildcards. The 'x' wildcard is used to represent single digits, e.g., 10.8.8.xx represents all addresses between 10.8.8.10 to 10.8.8.99. The '*' wildcard represents any number between 0 and 255, e.g., 10.8.8.* represents all addresses between 10.8.8.0 and 10.8.8.255. ▪ If the string 'ENUM' is specified for the destination IP address, an ENUM query containing the destination phone number is sent to the DNS server. The ENUM reply includes a SIP URI used as the Request-URI in the outgoing INVITE and for routing (if Proxy is not used). ▪ For detailed information on this feature and for configuring the 'Tel to IP Routing' table using the Web interface, refer to "Tel to IP Routing Table" on page 200. ▪ For available notations, refer to "Dialing Plan Notation" on page 194. ▪ For a description on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
PSTNPrefix	This *ini* file table parameter configures the routing of IP-to-Tel calls to Trunk Groups. The format of this parameter is as follows: [PSTNPrefix] FORMAT PstnPrefix_Index = PstnPrefix_**DestPrefix**, PstnPrefix_**TrunkGroupId**, PstnPrefix_**SourcePrefix,** PstnPrefix_**SourceAddress**, PstnPrefix_**ProfileId**, PstnPrefix_**SrcIPGroupID**, PstnPrefix_**DestHostPrefix**, PstnPrefix_**SrcHostPrefix**; [\PSTNPrefix] Where, ▪ DestPrefix = Destination number prefix. ▪ TrunkGroupId = Trunk Group ID (1-99). ▪ SourcePrefix = Source number prefix. ▪ SourceAddress = Source IP address (obtained from the Contact header in the INVITE message). ▪ ProfileId = Optional IP Profile ID (1-4) that can be applied to each routing rule. ▪ SrcIPGroupID = Source IP Group (1-9) associated with the incoming IP-to-Tel call.

ini File Parameter	Description
	▪ DestHostPrefix = Request URI host name prefix of the incoming INVITE message. ▪ SrcHostPrefix = From URI host name prefix of the incoming INVITE message. For example: [PSTNPrefix] FORMAT PstnPrefix_Index = PstnPrefix_DestPrefix, PstnPrefix_TrunkGroupId, PstnPrefix_SourcePrefix, PstnPrefix_SourceAddress, PstnPrefix_ProfileId, PstnPrefix_SrcIPGroupID, PstnPrefix_DestHostPrefix, PstnPrefix_SrcHostPrefix; PstnPrefix 0 = 100, 1, 200, *, 0, 2, , ; PstnPrefix 1 = *, 2, *, , 1, 3, acl, joe; [\PSTNPrefix] **Notes:** ▪ This parameter can include up to 24 indices. ▪ To support the In-Call Alternative Routing feature, you can use two entries that support the same call, but assigned with a different Trunk Group. The second entry functions as an alternative selection if the first rule fails as a result of one of the release reasons listed in the AltRouteCauseIP2Tel table. ▪ Selection of Trunk Groups (for IP-to-Tel calls) is according to destination number, source number,and source IP address. ▪ The source IP address (SourceAddress) can include the 'x' wildcard to represent single digits. For example: 10.8.8.xx represents all IP addresses between 10.8.8.10 and 10.8.8.99. ▪ The source IP address (SourceAddress) can include the asterisk ('*') wildcard to represent any number between 0 and 255. For example, 10.8.8.* represents all addresses between 10.8.8.0 and 10.8.8.255. ▪ If the source IP address (SourceAddress) includes an FQDN, DNS resolution is performed according to DNSQueryType. ▪ To configure IP-to-Tel routing using the Web interface, refer to "IP to Trunk Group Routing" on page 204. ▪ For available notations that represent multiple numbers, refer to "Dialing Plan Notation" on page 194. ▪ For a description on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
RemovePrefix	For a description of this parameter, refer to "Routing General Parameters" on page 198.
RouteModeIP2Tel	For a description of this parameter, refer to "IP to Trunk Group Routing" on page 204.
RouteModeTel2IP	For a description of this parameter, refer to "Tel to IP Routing Table" on page 200.
SwapRedirectNumber	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.
Prefix2RedirectNumber	For a description of this parameter, refer to "Configuring the Digital Gateway Parameters" on page 235.

ini File Parameter	Description
SwapTel2IPCalled&CallingNumbers	If enabled, the device swaps the calling and called numbers received from the Tel side. The INVITE message contains the swapped numbers. Applicable for Tel-to-IP calls. ▪ **[0]** = Disabled (default) ▪ **[1]** = Swap calling and called numbers
AddTON2RPI	For a description of this parameter, refer to "SIP General Parameters" on page 134.
NumberMapTel2IP	This *ini* file table parameter manipulates manipulates the destination number of Tel-to-IP calls. The format of this parameter is as follows: [NumberMapTel2Ip] FORMAT NumberMapTel2Ip_Index = NumberMapTel2Ip_**DestinationPrefix**, NumberMapTel2Ip_**SourcePrefix**, NumberMapTel2Ip_**SourceAddress**, NumberMapTel2Ip_**NumberType**, NumberMapTel2Ip_**NumberPlan**, NumberMapTel2Ip_**RemoveFromLeft**, NumberMapTel2Ip_**RemoveFromRight**, NumberMapTel2Ip_**LeaveFromRight**, NumberMapTel2Ip_**Prefix2Add**, NumberMapTel2Ip_**Suffix2Add**, NumberMapTel2Ip_**IsPresentationRestricted**; [\NumberMapTel2Ip] Where, ▪ DestinationPrefix = Destination number prefix. ▪ SourcePrefix = Source number prefix. ▪ SourceAddress = **N/A.** ▪ NumberType = Number Type used in RPID header. ▪ NumberPlan = Number Type used in RPID header. ▪ RemoveFromLeft = Number of stripped digits from the left. ▪ RemoveFromRight = Number of stripped digits from the right. ▪ LeaveFromRight = Number of remaining digits from the right. ▪ Prefix2Add = String to add as prefix. ▪ Suffix2Add = String to add as suffix. ▪ IsPresentationRestricted = N/A (set to $$). For example: [NumberMapTel2Ip] NumberMapTel2Ip 0 = 01,$$,*,0,0,2,$$,$$,971,$$,$$; NumberMapTel2Ip 1 = 10,10,*,255,255,3,0,5,100,$$,255; [\NumberMapTel2Ip] **Notes:** ▪ RemoveFromLeft, RemoveFromRight, Prefix2Add, Suffix2Add, LeaveFromRight, NumberType, and NumberPlan are applied if the called and calling numbers match the DestinationPrefix and SourcePrefix conditions. ▪ The manipulation rules are executed in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and Suffix2Add. ▪ Parameters can be skipped by using two dollar signs ('$$'). ▪ Number Plan and Type can optionally be used in Remote Party ID

ini File Parameter	Description
	(RPID) header by using the EnableRPIHeader and AddTON2RPI parameters. ▪ To configure manipulation of destination numbers for Tel-to-IP calls using the Web interface (i.e., in the 'Destination Phone Number Manipulation Table for Tel-to-IP Calls' table), refer to "Configuring the Number Manipulation Tables" on page 190). ▪ For a description on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
NumberMapIP2Tel	This *ini* file table parameter manipulates the destination number of IP-to-Tel calls. The format of this parameter is as follows: [NumberMapIp2Tel] FORMAT NumberMapIp2Tel_**Index** = NumberMapIp2Tel_**DestinationPrefix**, NumberMapIp2Tel_**SourcePrefix**, NumberMapIp2Tel_**SourceAddress**, NumberMapIp2Tel_**NumberType**, NumberMapIp2Tel_**NumberPlan**, NumberMapIp2Tel_**RemoveFromLeft**, NumberMapIp2Tel_**RemoveFromRight**, NumberMapIp2Tel_**LeaveFromRight**, NumberMapIp2Tel_**Prefix2Add**, NumberMapIp2Tel_**Suffix2Add**, NumberMapIp2Tel_**IsPresentationRestricted**; [\NumberMapIp2Tel] Where, ▪ DestinationPrefix = Destination number prefix. ▪ SourcePrefix = Source number prefix. ▪ SourceAddress = Source IP address (obtained from the Contact header in the INVITE message). ▪ NumberType = Q.931 Number Type (TON). ▪ NumberPlan = Q.931 Number Plan (NPI). ▪ RemoveFromLeft = Number of stripped digits from the left. ▪ RemoveFromRight = Number of stripped digits from the right. ▪ LeaveFromRight = Number of remaining digits from the right. ▪ Prefix2Add = String to add as prefix. ▪ Suffix2Add = String to add as suffix. ▪ IsPresentationRestricted = N/A (set to $$). For example: [NumberMapIp2Tel] NumberMapIp2Tel 0 = 01,034,10.13.77.8,$$,0,$$,2,$$,667,$$; NumberMapIp2Tel 1 = 10,10,1.1.1.1,255,255,3,0,5,100,$$,255; [\NumberMapIp2Tel] **Notes:** ▪ RemoveFromLeft, RemoveFromRight, Prefix2Add, Suffix2Add, LeaveFromRight, NumberType, and NumberPlan are applied if the called and calling numbers match the DestinationPrefix, SourcePrefix, and SourceAddress conditions. ▪ The manipulation rules are executed in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and Suffix2Add. ▪ Parameters can be skipped using two dollar signs ('$$').

ini File Parameter	Description
	▪ The Source IP address can include the 'x' wildcard to represent single digits. For example: 10.8.8.xx represents all addresses between 10.8.8.10 and 10.8.8.99. ▪ The Source IP address can include the asterisk ('*') wildcard to represent any number between 0 and 255. For example, 10.8.8.* represents all the addresses between 10.8.8.0 and 10.8.8.255. ▪ To configure manipulation of destination numbers for IP-to-Tel calls using the Web interface (i.e., in the 'Destination Phone Number Manipulation Table for IP-to-Tel Calls' table), refer to "Configuring the Number Manipulation Tables" on page 190). ▪ For a description on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
SourceNumberMapTel2IP	This *ini* file table parameter manipulates the source phone number for Tel-to-IP calls. The format of this parameter is as follows: [SourceNumberMapTel2Ip] FORMAT SourceNumberMapTel2Ip_**Index** = SourceNumberMapTel2Ip_**DestinationPrefix**, SourceNumberMapTel2Ip_**SourcePrefix**, SourceNumberMapTel2Ip_**SourceAddress**, SourceNumberMapTel2Ip_**NumberType**, SourceNumberMapTel2Ip_**NumberPlan**, SourceNumberMapTel2Ip_**RemoveFromLeft**, SourceNumberMapTel2Ip_**RemoveFromRight**, SourceNumberMapTel2Ip_**LeaveFromRight**, SourceNumberMapTel2Ip_**Prefix2Add**, SourceNumberMapTel2Ip_**Suffix2Add**, SourceNumberMapTel2Ip_**IsPresentationRestricted**; [\SourceNumberMapTel2Ip] Where, ▪ DestinationPrefix = Destination number prefix. ▪ SourcePrefix = Source number prefix. ▪ SourceAddress = Source IP address (obtained from the Request-URI in the INVITE message). ▪ NumberType = Number Type used in RPID header. ▪ NumberPlan = Number Plan used in RPID header. ▪ RemoveFromLeft = Number of stripped digits from the left. ▪ RemoveFromRight = Number of stripped digits from the right. ▪ LeaveFromRight = Number of remaining digits from the right. ▪ Prefix2Add = String to add as prefix. ▪ Suffix2Add = String to add as suffix. ▪ IsPresentationRestricted = Calling number presentation (0 to allow presentation; 1 to restrict presentation). For example: [SourceNumberMapTel2Ip] SourceNumberMapTel2Ip 0 = 22,03,$$,0,0,$$,2,$$,667,$$,0; SourceNumberMapTel2Ip 0 = 10,10,*,255,255,3,0,5,100,$$,255; [\SourceNumberMapTel2Ip]

ini File Parameter	Description
	Notes: ▪ RemoveFromLeft, RemoveFromRight, Prefix2Add, Suffix2Add, LeaveFromRight, NumberType, NumberPlan, and IsPresentationRestricted are applied if the called and calling numbers match the DestinationPrefix and SourcePrefix conditions. ▪ The manipulation rules are executed in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and Suffix2Add. ▪ Parameters can be skipped by using two dollar signs ('$$'). ▪ An asterisk ('*') represents all IP addresses. ▪ IsPresentationRestricted is set to 'Restricted' only if 'Asserted Identity Mode' is set to 'P-Asserted'. ▪ Number Plan and Type can optionally be used in Remote Party ID (RPID) header by using the EnableRPIHeader and AddTON2RPI parameters. ▪ To configure manipulation of source numbers for Tel-to-IP calls using the Web interface (i.e., in the 'Source Phone Number Manipulation Table for Tel-to-IP Calls' table), refer to "Configuring the Number Manipulation Tables" on page 190). ▪ For a description on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
SourceNumberMapIP2Tel	This *ini* file table parameter manipulates the source number for IP-to-Tel calls. The format of this parameter is as follows: [SourceNumberMapIp2Tel] FORMAT SourceNumberMapIp2Tel_**Index** = SourceNumberMapIp2Tel_**DestinationPrefix**, SourceNumberMapIp2Tel_**SourcePrefix**, SourceNumberMapIp2Tel_**SourceAddress**, SourceNumberMapIp2Tel_**NumberType**, SourceNumberMapIp2Tel_**NumberPlan**, SourceNumberMapIp2Tel_**RemoveFromLeft**, SourceNumberMapIp2Tel_**RemoveFromRight**, SourceNumberMapIp2Tel_**LeaveFromRight**, SourceNumberMapIp2Tel_**Prefix2Add**, SourceNumberMapIp2Tel_**Suffix2Add**, SourceNumberMapIp2Tel_**IsPresentationRestricted**; [\SourceNumberMapIp2Tel] Where, ▪ DestinationPrefix = Destination number prefix. ▪ SourcePrefix = Source number prefix. ▪ SourceAddress = Source IP address (obtained from the Request-URI in the INVITE message). ▪ NumberType = Q.931 Number Type (TON). ▪ NumberPlan = Q.931 Number Plan (NPI). ▪ RemoveFromLeft = Number of stripped digits from the left. ▪ RemoveFromRight = Number of stripped digits from the right. ▪ LeaveFromRight = Number of remaining digits from the right. ▪ Prefix2Add = String to add as prefix. ▪ Suffix2Add = String to add as suffix.

ini File Parameter	Description
	▪ IsPresentationRestricted = Calling number presentation (0 to allow presentation; 1 to restrict presentation). For example: [SourceNumberMapIp2Tel] SourceNumberMapIp2Tel 0 = 22,03,$$,$$,$$,$$,2,667,$$,$$;SourceNumberMapIp2Tel 1 = 034,01,1.1.1.1,$$,0,2,$$,$$,972,$$,10; [\SourceNumberMapIp2Tel] **Notes:** ▪ RemoveFromLeft, RemoveFromRight, Prefix2Add, Suffix2Add, LeaveFromRight, NumberType, and NumberPlan are applied if the called and calling numbers match the DestinationPrefix, SourcePrefix, and SourceAddress conditions. ▪ The manipulation rules are executed in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and Suffix2Add. ▪ Parameters can be skipped by using two dollar signs ('$$'). ▪ The Source IP address can include the 'x' wildcard to represent single digits. For example: 10.8.8.xx represents all addresses between 10.8.8.10 and 10.8.8.99. ▪ The Source IP address can include the asterisk ('*') wildcard to represent any number between 0 and 255. For example, 10.8.8.* represents all the addresses between 10.8.8.0 and 10.8.8.255. ▪ To configure manipulation of source numbers for IP-to-Tel calls using the Web interface (i.e., in the 'Source Phone Number Manipulation Table for IP-to-Tel Calls' table), refer to "Configuring the Number Manipulation Tables" on page 190). ▪ For a description on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.

ini File Parameter	Description
For ETSI ISDN variant, the following Number Plan and Type combinations (Plan/Type) are supported in the Destination and Source Manipulation tables: ▪ 0,0 = Unknown, Unknown ▪ 9,0 = Private, Unknown ▪ 9,1 = Private, Level 2 Regional ▪ 9,2 = Private, Level 1 Regional ▪ 9,3 = Private, PISN Specific ▪ 9,4 = Private, Level 0 Regional (local) ▪ 1,0 = Public(ISDN/E.164), Unknown ▪ 1,1 = Public(ISDN/E.164), International ▪ 1,2 = Public(ISDN/E.164), National ▪ 1,3 = Public(ISDN/E.164), Network Specific ▪ 1,4 = Public(ISDN/E.164), Subscriber ▪ 1,6 = Public(ISDN/E.164), Abbreviated For NI-2 and DMS-100 ISDN variants the valid combinations of TON and NPI for calling and called numbers are (Plan/Type): ▪ 0/0 - Unknown/Unknown ▪ 1/1 - International number in ISDN/Telephony numbering plan ▪ 1/2 - National number in ISDN/Telephony numbering plan ▪ 1/4 - Subscriber (local) number in ISDN/Telephony numbering plan ▪ 9/4 - Subscriber (local) number in Private numbering plan	
SecureCallsFromIP	For a description of this parameter, refer to "Advanced Parameters" on page 166.
AltRouteCauseTel2IP	This *ini* file table parameter configures SIP call failure reason values received from the IP side. If a call is released as a result of one of these reasons, the device attempts to locate an alternative route to the call in the 'Tel to IP Routing' table (if Proxy is not used) or used as a redundant Proxy (when Proxy is used). The format of this parameter is as follows: [AltRouteCauseTel2IP] FORMAT AltRouteCauseTel2IP_**Index** = AltRouteCauseTel2IP_**ReleaseCause**; [\AltRouteCauseTel2IP] For example: [AltRouteCauseTel2IP] AltRouteCauseTel2IP 0 = 486; (Busy Here) AltRouteCauseTel2IP 1 = 480; (Temporarily Unavailable) AltRouteCauseTel2IP 2 = 408; (No Response) [\AltRouteCauseTel2IP] **Notes:** ▪ The 408 reason can be used to specify no response from the remote party to the INVITE request. ▪ This parameter can include up to 5 indices. ▪ For defining the Reasons for Alternative Routing table using the Web interface, refer to "Reasons for Alternative Routing" on page 209. ▪ For an explanation on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.

ini File Parameter	Description
AltRouteCauseIP2Tel	This *ini* file table parameter configures call failure reason values received from the PSTN side (in Q.931 presentation). If a call is released as a result of one of these reasons, the device attempts to locate an alternative Trunk Group for the call in the 'IP to Trunk Group Routing' table. The format of this parameter is as follows: [AltRouteCauseIP2Tel] FORMAT AltRouteCauseIP2Tel_**Index** = AltRouteCauseIP2Tel_**ReleaseCause**; [\AltRouteCauseIP2Tel] For example: [AltRouteCauseIP2Tel] AltRouteCauseIP2Tel 0 = 3 (No Route to Destination) AltRouteCauseIP2Tel 1 = 1 (Unallocated Number) AltRouteCauseIP2Tel 2 = 17 (Busy Here) [\AltRouteCauseIP2Tel] **Notes:** ▪ This parameter can include up to 5 indices. ▪ If the device fails to establish a call to the PSTN because it has no available channels in a specific trunk group (e.g., all trunk group's channels are occupied, or the trunk group's spans are disconnected or out of sync), it uses the Internal Release Cause '3' (No Route to Destination). This cause can be used in the AltRouteCauseIP2Tel table to define routing to an alternative trunk group. ▪ For defining the Reasons for Alternative Routing table using the Web interface, refer to "Reasons for Alternative Routing" on page 209. ▪ For an explanation on usng *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.
EnableETSIDiversion	Defines the method in which the Redirect Number is passed toward the Tel side. ▪ **[0]** = Q.931 Redirecting Number Information Element (IE) (default) ▪ **[1]** = ETSI DivertingLegInformation2 in a Facility IE
CopyDest2RedirectNumber	Determines whether the device copies the Called Number to the Redirect Number for Tel-to-IP calls. ▪ **[0]** Disable (default). ▪ **[1]** Enable. **Note:** If a Redirect Number already appears in the incoming call, it is overridden by the Called Number.
FilterCalls2IP	For a description of this parameter, refer to "Advanced Parameters" on page 166.

ini File Parameter	Description
Alternative Routing Parameters	
RedundantRoutingMode	Determines the type of redundant routing mechanism to implement when a call can't be completed using the main route. ▪ **[0]** = No redundant routing is used. If the call can't be completed using the main route (using the active Proxy or the first matching rule in the internal routing table), the call is disconnected. ▪ **[1]** = Internal routing table is used to locate a redundant route (default). ▪ **[2]** = Proxy list is used to locate a redundant route.
AltRoutingTel2IPEnable	For a description of this parameter, refer to "Routing General Parameters" on page 198.
AltRoutingTel2IPMode	For a description of this parameter, refer to "Routing General Parameters" on page 198.
AltRoutingTel2IPConnMethod	For a description of this parameter, refer to "Routing General Parameters" on page 198.
IPConnQoSMaxAllowedPL	For a description of this parameter, refer to "Routing General Parameters" on page 198.
IPConnQoSMaxAllowedDelay	For a description of this parameter, refer to "Routing General Parameters" on page 198.
Phone-Context Parameters	
AddPhoneContextAsPrefix	For a description of this parameter, refer to "Mapping NPI/TON to Phone-Context" on page 195.
PhoneContext	This *ini* file table parameter defines the Phone Context table. The format for this parameter is as follows: [PhoneContext] FORMAT PhoneContext_**Index** = PhoneContext_**Npi**, PhoneContext_**Ton**, PhoneContext_**Context**; [\PhoneContext] Where, ▪ Npi = Number Plan. ▪ Ton = Type of Number. ▪ Context = Phone-Context value. When a call is received from the ISDN/Tel, the NPI and TON are compared to the table, and the Phone-Context value is used in the outgoing SIP INVITE message. The same mapping occurs when an INVITE with a Phone-Context attribute is received. The Phone-Context parameter appears in the standard SIP headers where a phone number is used (Request-URI, To, From, Diversion). For example: [PhoneContext] PhoneContext 0 = 0,0,unknown.com PhoneContext 1 = 1,1,host.com PhoneContext 2 = 9,1,na.e164.host.com [\PhoneContext] **Notes:** ▪ This parameter can include up to 20 indices.

ini File Parameter	Description
	▪ Several entries with the same NPI-TON or Phone-Context are allowed. In this scenario, a Tel-to-IP call uses the first match. ▪ Phone-Context '+' is a unique as it doesn't appear in the Request-URI as a Phone-Context parameter. Instead, it's added as a prefix to the phone number. The '+' isn't removed from the phone number in the IP-to-Tel direction. ▪ To configure the Phone Context table using the Web interface, refer to "Mapping NPI/TON to Phone-Context" on page 195. ▪ For a description on using *ini* file table parameters, refer to "Structure of ini File Table Parameters" on page 293.

4.4.14 Channel Parameters

The Channel Parameters define the DTMF, fax and modem transfer modes.

Table 4-14: Channel Parameters

ini File Parameter	Description
DJBufMinDelay	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88.
DJBufOptFactor	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88.
AnalogSignalTransportType	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88. **Note:** This parameter is applicable only to Mediant 1000.
FaxTransportMode	For a description of this parameter, refer to "Configuring the Fax / Modem / CID Settings" on page 84.
FaxRelayEnhancedRedundancyDepth	For a description of this parameter, refer to "Configuring the Fax / Modem / CID Settings" on page 84.
FaxRelayRedundancyDepth	For a description of this parameter, refer to "Configuring the Fax / Modem / CID Settings" on page 84.
FaxRelayMaxRate	For a description of this parameter, refer to "Configuring the Fax / Modem / CID Settings" on page 84.
FaxRelayECMEnable	For a description of this parameter, refer to "Configuring the Fax / Modem / CID Settings" on page 84.
FaxModemBypassCoderType	For a description of this parameter, refer to "Configuring the Fax / Modem / CID Settings" on page 84.
CNGDetectorMode	For a description of this parameter, refer to "Configuring the Fax / Modem / CID Settings" on page 84.
FaxCNGMode	Determines the device's behavior upon detection of a CNG tone. ▪ **[0]** = Does not send a SIP Re-INVITE upon detection of a fax CNG tone when CNGDetectorMode is set to 1 (default). ▪ **[1]** = Sends a SIP Re-INVITE upon detection of a fax CNG tone when CNGDetectorMode is set to 1.

ini File Parameter	Description
FaxModemBypassM	For a description of this parameter, refer to "Configuring the Fax / Modem / CID Settings" on page 84.
FaxBypassPayloadType	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88.
CallerIDTransportType	For a description of this parameter, refer to "Configuring the Fax / Modem / CID Settings" on page 84.
ModemBypassPayloadType	Modem Bypass dynamic payload type. The range is 0-127. The default value is 103.
FaxModemRelayVolume	Determines the fax gain control. The range -18 to -3 corresponds to -18 dBm to -3 dBm in 1-dB steps. The default is -6 dBm fax gain control. **Note:** This parameter is applicable only to Mediant 1000.
DetFaxOnAnswerTone	For a description of this parameter, refer to "SIP General Parameters" on page 134.
NTEMaxDuration	Maximum time for sending Named Telephony Events (NTEs) to the IP side, regardless of the time range when the TDM signal is detected. The range is -1 to 200,000,000 msec (i.e., 55 hours). The default is -1 (i.e., NTE stops only upon detection of an End event).
EchoCancellerAggressiveNLP	Enables or disables the Aggressive NLP at the first 0.5 second of the call. When enabled, the echo is removed only in the first half a second of the incoming IP signal. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
FaxModemBypassBasicRTPPacketInterval	Determines the basic frame size that is used during fax / modem bypass sessions. ▪ **[0]** = Determined internally (default) ▪ **[1]** = 5 msec (not recommended) ▪ **[2]** = 10 msec ▪ **[3]** = 20 msec **Note:** When set for 5 msec (1), the maximum number of simultaneous channels supported is 120.
FaxModemBypassDJBufMinDelay	Determines the Jitter Buffer delay (in milliseconds) during fax and modem bypass session. The range is 0 to 150 msec. The default is 40.
EnableFaxModemInbandNetworkDetection	Enables or disables inband network detection related to fax/modem. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable When this parameter is enabled on Bypass mode (VxxTransportType = 2), a detection of an Answer Tone from the network triggers a switch to bypass mode in addition to the local Fax/Modem tone detections. However, only a high bit-rate coder voice session effectively detects the Answer Tone sent by a remote Endpoint. This can be useful when, for example, the payload of voice and bypass is the same, allowing the originator to switch to bypass mode as well.

ini File Parameter	Description
NSEMode	Cisco compatible fax and modem bypass mode. ▪ **[0]** = NSE disabled (default) ▪ **[1]** = NSE enabled **Notes:** ▪ This feature can be used only if VxxModemTransportType = 2 (Bypass). ▪ If NSE mode is enabled, the SDP contains the following line: 'a=rtpmap:100 X-NSE/8000'. ▪ To use this feature: -- The Cisco gateway must include the following definition: 'modem passthrough nse payload-type 100 codec g711alaw'. -- Set the Modem transport type to Bypass mode (VxxModemTransportType = 2) for all modems. -- Configure the gateway parameter NSEPayloadType = 100. In NSE bypass mode, the device starts using G.711 A-Law (default) or G.711μ-Law according to the parameter FaxModemBypassCoderType. The payload type used with these G.711 coders is a standard one (8 for G.711 A-Law and 0 for G.711 μ-Law). The parameters defining payload type for the 'old' AudioCodes' Bypass mode FaxBypassPayloadType and ModemBypassPayloadType are not used with NSE Bypass. The bypass packet interval is selected according to the parameter FaxModemBypassBasicRtpPacketInterval.
NSEPayloadType	NSE payload type for Cisco Bypass compatible mode. The valid range is 96-127. The default value is 105. **Note:** Cisco gateways usually use NSE payload type of 100.
V21ModemTransportType	For a description of this parameter, refer to "Configuring the Fax / Modem / CID Settings" on page 84.
V22ModemTransportType	For a description of this parameter, refer to "Configuring the Fax / Modem / CID Settings" on page 84.
V23ModemTransportType	For a description of this parameter, refer to "Configuring the Fax / Modem / CID Settings" on page 84.
V32ModemTransportType	For a description of this parameter, refer to "Configuring the Fax / Modem / CID Settings" on page 84.
V34ModemTransportType	For a description of this parameter, refer to "Configuring the Fax / Modem / CID Settings" on page 84.
BellModemTransportType	Determines the Bell modem transport method. ▪ **[0]** = Transparent (default). ▪ **[2]** = Bypass. ▪ **[3]** = Transparent with events.
InputGain	For a description of this parameter, refer to "Configuring the Voice Settings" on page 81.
VoiceVolume	For a description of this parameter, refer to "Configuring the Voice Settings" on page 81.
RTPRedundancyDepth	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88.

ini File Parameter	Description
RFC2198PayloadType	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88.
EnableSilenceCompression	For a description of this parameter, refer to "Configuring the Voice Settings" on page 81.
IsCiscoSCEMode	Determines whether a Cisco gateway exists at the remote side. ▪ **[0]** = No Cisco gateway exists at the remote side (default). ▪ **[1]** = A Cisco gateway exists at the remote side. When there is a Cisco gateway at the remote side, the device must set the value of the 'annexb' parameter of the fmtp attribute in the SDP to 'no'. This logic is used if EnableSilenceCompression = 2 (enable without adaptation). In this case, Silence Suppression is used on the channel, but not declared in the SDP. **Note:** The IsCiscoSCEMode parameter is only relevant when the selected coder is G.729.
EnableEchoCanceller	For a description of this parameter, refer to "Configuring the Voice Settings" on page 81.
MaxEchoCancellerLength	For a description of this parameter, refer to Configuring the General Media Settings on page 93.
EchoCancellerAggressiveNLP	Enables or disables the Aggressive Non-Linear Processor (NLP) in the first 0.5 second of the call. ▪ **[0]** = Disabled (default) ▪ **[1]** = Enabled
EnableNoiseReduction	Enables / disables the DSP Noise Reduction mechanism. ▪ **[0]** = Disable (default). ▪ **[1]** = Enable. **Note:** When this parameter is enabled the channel capacity might be reduced.
TestMode	Defines the type of testing mode: ▪ **[0]** = CoderLoopback, encoder-decoder loopback inside DSP. ▪ **[1]** = PCMLoopback, loopback the incoming PCM to the outgoing PCM. ▪ **[2]** = ToneInjection, generates a 1000-Hz tone to outgoing PCM. ▪ **[3]** = NoLoopback, (default). **Note:** This parameter is applicable only to Mediant 1000.
EnableStandardSIDPayloadType	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88.
ComfortNoiseNegotiation	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88.
RTPSIDCoeffNum	Determines the number of spectral coefficients added to an SID packet being sent according to RFC 3389. Valid only if EnableStandardSIDPayloadType is set to 1. The valid values are **[0]** (default), **[4]**, **[6]**, **[8]** and **[10]**.
DTMFVolume	For a description of this parameter, refer to "Configuring the Voice Settings" on page 81.

ini File Parameter	Description
DTMFGenerationTwist	For a description of this parameter, refer to "Configuring the Voice Settings" on page 81.
DTMFInterDigitInterval	Time in msec between generated DTMF digits to PSTN side (if TxDTMFOption = 1, 2 or 3). The default value is 100 msec. The valid range is 0 to 32767.
DTMFDigitLength	Time (in msec) for generating DTMF tones to the PSTN side (if TxDTMFOption = 1, 2 or 3). It also configures the duration that is sent in INFO (Cisco) messages. The valid range is 0 to 32767. The default value is 100.
RxDTMFHangOverTime	Defines the Voice Silence time (in msec units) after playing DTMF or MF digits to the Tel / PSTN side that arrive as Relay from the IP side. Valid range is 0 to 2,000 msec. The default is 1,000 msec.
TxDTMFHangOverTime	Defines the Voice Silence time (in msec units) after detecting the end of DTMF or MF digits at the Tel / PSTN side when the DTMF Transport Type is either Relay or Mute. Valid range is 0 to 2,000 msec. The default is 1,000 msec.
DTMFTransportType	For a description of this parameter, refer to "Configuring the Voice Settings" on page 81.
AnswerDetectorSensitivity	For a description of this parameter, refer to Configuring the Voice Settings on page 81. **Note:** This parameter is applicable only to Mediant 1000.
RFC2833PayloadType	For a description of this parameter, refer to "DTMF & Dialing Parameters" on page 160.
UserDefinedToneDetectorEnable	Enables or disables detection of User Defined Tones signaling. ▪ **[0]** = Disable ▪ **[1]** = Enable
UDTDetectorFrequencyDeviation	Defines the deviation (in Hz) allowed for the detection of each signal frequency. The valid range is 1 to 50. The default value is 50.
CPTDetectorFrequencyDeviation	Defines the deviation (in Hz) allowed for the detection of each CPT signal frequency. The valid range is 1 to 30. The default value is 10.
MGCPDTMFDetectionPoint	▪ **[0]** = DTMF event is reported on the end of a detected DTMF digit. ▪ **[1]** = DTMF event is reported on the start of a detected DTMF digit (default).

ini File Parameter	Description
KeyBlindTransfer	Keypad sequence that activates blind transfer for Tel-to-IP calls. There are two possible scenarios: ▪ Option 1: After this sequence is dialed, the current call is put on hold (using Re-INVITE), a dial tone is played to the B-channel, and then phone number collection starts. ▪ Option 2: A Hook-Flash is pressed, the current call is put on hold, a dial tone is played to the B-channel, and then digit collection starts. After this sequence is identified, the device continues the collection of the destination phone number. For both options, after the phone number is collected, it's sent to the transferee in a SIP REFER request (without a Replaces header). The call is then terminated and a confirmation tone is played to the B-channel. If the phone number collection fails due to a mismatch, a reorder tone is played to the B-channel. **Note:** It is possible to configure whether the KeyBlindTransfer code is added as a prefix to the dialed destination number, by using the parameter KeyBlindTransferAddPrefix.
KeyBlindTransferAddPrefix	Determines whether the device adds the Blind Transfer code (KeyBlindTransfer) to the dialed destination number. ▪ **[0]** Disable (default). ▪ **[1]** Enable. **Note:** This parameter is applicable to FXO and FXS interfaces.
MinFlashHookTime	For a description of this parameter, refer to "Configuring the Hook-Flash Settings" on page 94. **Note:** This parameter is applicable only to Mediant 1000.
FlashHookOption	For a description of this parameter, refer to DTMF & Dialing Parameters on page 160. **Note:** This parameter is applicable only to Mediant 1000.
FlashHookPeriod	For a description of this parameter, refer to "Configuring the Hook-Flash Settings" on page 94. **Note:** This parameter is applicable only to Mediant 1000.
VQMonEnable	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88.
RTCPInterval	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88.
DisableRTCPRandomize	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88.
RTCPXRESCTransportType	Determines the transport layer used for outgoing SIP dialogs initiated by the device to the RTCP-XR Collection Server. ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When set to 'Not Configured', the value of the parameter SIPTransportType is used.

ini File Parameter	Description
RTCPXREscIP	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88.
RTCPXRReportMode	For a description of this parameter, refer to "Configuring the RTP / RTCP Settings" on page 88.

4.4.15 Auxiliary / Configuration Files Parameters

The configuration files (i.e., auxiliary files) can be loaded to the device using the Web interface or a TFTP session (refer to "Auxiliary Files" on page 266). Before you load them to the device, you need to specify these files in the *ini* file and whether they must be stored in the non-volatile memory. The table below lists the *ini* file parameters associated with these auxiliary files:

Table 4-15: Auxiliary / Configuration Files Parameters

ini File Parameter	Description
CallProgressTonesFilename	The name of the file containing the Call Progress Tones definitions. Refer to the *Product Reference Manual* for additional information on how to create and load this file.
FXSLoopCharacteristicsFileName	The name (and path) of the file providing the FXS line characteristic parameters. **Note:** This parameter is applicable only to Mediant 1000.
FXOLoopCharacteristicsFileName	The name (and path) of the file providing the FXO line characteristic parameters. **Note:** This parameter is applicable only to Mediant 1000.
CASFileName	This is the name of the file containing specific CAS protocol definition (such as 'E_M_WinkTable.dat'). These files are provided to support various types of CAS signaling.
CASFileName_x	CAS file name (e.g., 'E_M_WinkTable.dat') that defines the CAS protocol. It is possible to define up to eight different CAS files by repeating this parameter. Each CAS file can be associated with one or more of the device trunks using the parameter CASTableIndex_x.
CASTablesNum	Number 1 to 8. Specifies how many CAS configuration files are loaded.
VoicePromptsFileName	The name (and path) of the file containing the Voice Prompts definitions. Refer to the Product Reference Manual for additional information on how to create and load this file.
PrerecordedTonesFileName	The name (and path) of the file containing the Prerecorded Tones.
CasTrunkDialPlanName	The Dial Plan name (up to 11-character strings) that is used on the specific trunk.
DialPlanFileName	The name (and path) of the file containing dial-plan configuration for CAS and SIP protocols. This file should be constructed using the TrunkPack Conversion Utility (refer to the Product Reference Manual) supplied as part of the software package on the CD accompanying the device.

ini File Parameter	Description
UserInfoFileName	The name (and path) of the file containing the User Information data.
SaveConfiguration	Determines if the device's configuration (parameters and files) is saved to flash (non-volatile memory). ▪ **[0]** = Configuration isn't saved to flash memory. ▪ **[1]** = Configuration is saved to flash memory (default).

5 Default Settings

You can restore the device's factory default settings or define your own default settings for the device.

Note: You can also restore factory default settings using the device's hardware reset button (refer to the device's Fast Track Guide).

5.1 Defining Default Settings

The device is shipped with factory default configuration values stored on its non-volatile flash memory. However, you can define your own default values instead of using the factory defaults. This is performed using an*ini* file that includes the header '[ClientDefaults]'. Below this header, simply define new default values for the required *ini* file parameters. The parameters are defined in the same format as in the standard *ini* file, and loaded to the device using TFTP (i.e., not via the Web interface).

➢ To define default values for device parameters, take these 2 steps:

1. Configure the ClientDefaults *ini* file with new default parameter values, as needed.
2. Load the ClientDefaults *ini* file to the device using TFTP (refer to the *Product Reference Manual*).

An example of a ClientsDefault *ini* file for defining default values for Syslog server parameters is shown below:

```
[ClientDefaults]
EnableSyslog = 1
SyslogServerIP = 10.13.2.20
```

➢ To remove user-defined defaults and restore factory default values, take this step:

- Load an empty (i.e., without any parameters) ClientDefaults *ini* file to the device, using TFTP.

5.2 Restoring Factory Defaults

You can restore all default settings to th edevice using the *ini* file. When a parameter is absent from a loaded *ini* file, the default value is assigned to that parameter (according to the *cmp* file loaded to the device) and stored in the non-volatile memory (thereby, overriding the value previously defined for that parameter). Therefore, to restore the device's default configuration parameters, load an *ini* file without any parameters or with a semicolon (;) preceding all lines in the file.

Reader's Notes

6 Auxiliary Configuration Files

This section describes the auxiliary files (with the *dat* file extension), which are loaded, in addition to the *ini* file, to the device. You can load the auxiliary files to the device using one of the following methods:

- Web interface (refer to "Loading Auxiliary Files" on page 266)
- *ini* file: specify the name of the relevant auxiliary file in the device's *ini* file and then load the *ini* file to the device (refer to "Loading Auxiliary Files" on page 266).

6.1 Configuring the Call Progress Tones File

The Call Progress Tones (CPT) and Distinctive Ringing (applicable only to Mediant 1000) auxiliary file used by the device is a binary file (with file extension *dat*). This file is comprised of two sections. The first section contains the definitions of the Call Progress Tones (levels and frequencies) that are detected / generated by the device. The second section contains the characteristics of the distinctive ringing signals that are generated by the device (refer to Configuring the Distinctive Ringing Section of the ini File on page 390).

You can either use one of the supplied device auxiliary (*dat*) files or create your own file. To create your own auxiliary file, it's recommended to modify the supplied *usa_tone.ini* file (in any standard text editor) to suit your specific requirements, and to convert the modified *ini* file into binary format using the TrunkPack Downloadable Conversion Utility. For the description of the procedure on how to convert CPT *ini* file into a binary *dat* file, refer to the *Product Reference Manual*.

To load the Call Progress Tones *(dat*) file to the device, use the Web interface or *ini* file (refer to "Loading Auxiliary Files" on page 266).

Note: Only the *dat* file can be loaded to the device.

You can create up to 32 different Call Progress Tones, each with frequency and format attributes. The frequency attribute can be single or dual-frequency (in the range of 300 to 1980 Hz), or an Amplitude Modulated (AM). In total, up to 64 different frequencies are supported. Only eight AM tones, in the range of 1 to 128 kHz, can be configured (the detection range is limited to 1 to 50 kHz). Note that when a tone is composed of a single frequency, the second frequency field must be set to zero.

The format attribute can be one of the following:

- **Continuous:** (e.g., dial tone) a steady non-interrupted sound. Only the 'First Signal On time' should be specified. All other on and off periods must be set to zero. In this case, the parameter specifies the detection period. For example, if it equals 300, the tone is detected after 3 seconds (300 x 10 msec). The minimum detection time is 100 msec.
- **Cadence:** A repeating sequence of on and off sounds. Up to four different sets of on / off periods can be specified.
- **Burst:** A single sound followed by silence. Only the 'First Signal On time' and 'First Signal Off time' should be specified. All other on and off periods must be set to zero. The burst tone is detected after the off time is completed.

You can specify several tones of the same type. These additional tones are used only for tone detection. Generation of a specific tone conforms to the first definition of the specific tone. For example, you can define an additional dial tone by appending the second dial tone's definition lines to the first tone definition in the *ini* file. The device reports dial tone detection if either of the two tones is detected.

The Call Progress Tones section of the *ini* file comprises the following segments:

- **[NUMBER OF CALL PROGRESS TONES]:** Contains the following key:
  'Number of Call Progress Tones' defining the number of Call Progress Tones that are defined in the file.
- **[CALL PROGRESS TONE #X]:** containing the Xth tone definition (starting from 1 and not exceeding the number of Call Progress Tones defined in the first section) using the following keys:
  - **Tone Type:** Call Progress Tone types:
    - **[1]** Dial Tone
    - **[2]** Ringback Tone
    - **[3]** Busy Tone
    - **[7]** Reorder Tone
    - **[8]** Confirmation Tone (Applicable only to Analog devices)
    - **[9]** Call Waiting Tone (Applicable only to Analog devices)
    - **[15]** Stutter Dial Tone (Applicable only to Analog devices)
    - **[16]** Off Hook Warning Tone (Applicable only to Analog devices)
    - **[17]** Call Waiting Ringback Tone
    - **[23]** Hold Tone
  - **Tone Modulation Type:** Either Amplitude Modulated (1) or regular (0).
  - **Tone Form:** The tone's format can be one of the following:
    - Continuous (1)
    - Cadence (2)
    - Burst (3)
  - **Low Freq [Hz]:** frequency (in Hz) of the lower tone component in case of dual frequency tone, or the frequency of the tone in case of single tone. This is not relevant to Amplitude Modulated (AM) tones.
  - **High Freq [Hz:** frequency (in Hz) of the higher tone component in case of dual frequency tone, or zero (0) in case of single tone (not relevant to AM tones).
  - **Low Freq Level [-dBm]:** generation level 0 dBm to -31 dBm in dBm (not relevant to AM tones).
  - **High Freq Level:** generation level. 0 to -31 dBm. The value should be set to 32 in the case of a single tone (not relevant to AM tones).
  - **First Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the first cadence on-off cycle. For be continuous tones, this parameter defines the detection period. For burst tones, it defines the tone's duration.
  - **First Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the first cadence on-off cycle (for cadence tones). For burst tones, this parameter defines the off time required after the burst tone ends and the tone detection is reported. For continuous tones, this parameter is ignored.

- **Second Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.
- **Second Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.
- **Third Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the third cadence ON-OFF cycle. Can be omitted if there isn't a third cadence.
- **Third Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the third cadence ON-OFF cycle. Can be omitted if there isn't a third cadence.
- **Fourth Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the fourth cadence ON-OFF cycle. Can be omitted if there isn't a fourth cadence.
- **Fourth Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the fourth cadence ON-OFF cycle. Can be omitted if there isn't a fourth cadence.
- **Carrier Freq [Hz]:** frequency of the carrier signal for AM tones.
- **Modulation Freq [Hz]:** frequency of the modulated signal for AM tones (valid range from 1 to 128 Hz).
- **Signal Level [-dBm]:** level of the tone for AM tones.
- **AM Factor [steps of 0.02]:** amplitude modulation factor (valid range from 1 to 50. Recommended values from 10 to 25).

Notes:

- When the same frequency is used for a continuous tone and a cadence tone, the 'Signal On Time' parameter of the continuous tone must have a value that is greater than the 'Signal On Time' parameter of the cadence tone. Otherwise the continuous tone is detected instead of the cadence tone.
- The tones frequency should differ by at least 40 Hz from one tone to other defined tones.

For example, to configure the dial tone to 440 Hz only, enter the following text:

```
#Dial tone
[CALL PROGRESS TONE #1]
Tone Type=1
Tone Form =1 (continuous)
Low Freq [Hz]=440
High Freq [Hz]=0
Low Freq Level [-dBm]=10 (-10 dBm)
High Freq Level [-dBm]=32 (use 32 only if a single tone is
required)
First Signal On Time [10msec]=300; the dial tone is detected after
3 sec
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=0
Second Signal Off Time [10msec]=0
```

6.2 Configuring the Distinctive Ringing Section of the ini File

Distinctive Ringing is only applicable to FXS interface. Using the distinctive ringing section of this auxiliary file, you can create up to 16 distinctive ringing patterns. Each ringing pattern configures the ringing tone frequency and up to four ringing cadences. The same ringing frequency is used for all the ringing pattern cadences. The ringing frequency can be configured in the range of 10 to 200 Hz with a 5 Hz resolution.

Note: Distinctive Ringing is applicable only to Mediant 1000.

Each of the ringing pattern cadences is specified by the following parameters:

- **Burst Ring On Time:** Configures the cadence to be a burst cadence in the entire ringing pattern. The burst relates to On time and the Off time of the same cadence. It must appear between 'First/Second/Third/Fourth' string and the 'Ring On/Off Time' This cadence rings once during the ringing pattern. Otherwise, the cadence is interpreted as cyclic: it repeats for every ringing cycle.
- **Ring On Time:** specifies the duration of the ringing signal.
- **Ring Off Time:** specifies the silence period of the cadence.

The distinctive ringing section of the *ini* file format contains the following strings:

- **[NUMBER OF DISTINCTIVE RINGING PATTERNS]:** Contains the following key:
  - 'Number of Distinctive Ringing Patterns' defining the number of Distinctive Ringing signals that are defined in the file.
- **[Ringing Pattern #X]:** Contains the Xth ringing pattern definition (starting from 0 and not exceeding the number of Distinctive Ringing patterns defined in the first section minus 1) using the following keys:
  - **Ring Type:** Must be equal to the Ringing Pattern number.
  - **Freq [Hz]:** Frequency in hertz of the ringing tone.
  - **First (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the first cadence on-off cycle.
  - **First (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the first cadence on-off cycle.
  - **Second (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the second cadence on-off cycle.
  - **Second (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the second cadence on-off cycle.
  - **Third (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the third cadence on-off cycle.
  - **Third (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the third cadence on-off cycle.

- **Fourth (Burst) Ring On Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the fourth cadence on-off cycle.
- **Fourth (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the fourth cadence on-off cycle.

Note: In SIP, the distinctive ringing pattern is selected according to Alert-Info header that is included in the INVITE message. For example, Alert-Info <Bellcore-dr2>, or Alert-Info<http://…/Bellcore-dr2>. 'dr2' defines ringing pattern # 2. If the Alert-Info header is missing, the default ringing tone (0) is played.

6.2.1 Examples of Ringing Signals

- Below is an example of a **ringing burst**:

```
#Three ringing bursts followed by repeated ringing of 1 sec on and
3 sec off.
[NUMBER OF DISTINCTIVE RINGING PATTERNS]
Number of Ringing Patterns=1
[Ringing Pattern #0]
Ring Type=0
Freq [Hz]=25
First Burst Ring On Time [10msec]=30
First Burst Ring Off Time [10msec]=30
Second Burst Ring On Time [10msec]=30
Second Burst Ring Off Time [10msec]=30
Third Burst Ring On Time [10msec]=30
Third Burst Ring Off Time [10msec]=30
Fourth Ring On Time [10msec]=100
Fourth Ring Off Time [10msec]=300
```

- Below is an example of **various ringing signals**:

```
[NUMBER OF DISTINCTIVE RINGING PATTERNS]
Number of Ringing Patterns=3
#Regular North American Ringing Pattern
[Ringing Pattern #0]
Ring Type=0
Freq [Hz]=20
First Ring On Time [10msec]=200
First Ring Off Time [10msec]=400

#GR-506-CORE Ringing Pattern 1
[Ringing Pattern #1]
Ring Type=1
Freq [Hz]=20
First Ring On Time [10msec]=200
First Ring Off Time [10msec]=400

#GR-506-CORE Ringing Pattern 2
[Ringing Pattern #2]
Ring Type=2
Freq [Hz]=20
First Ring On Time [10msec]=80
First Ring Off Time [10msec]=40
Second Ring On Time [10msec]=80
Second Ring Off Time [10msec]=400
```

6.3 Prerecorded Tones (PRT) File

The Call Progress Tones (CPT) mechanism has several limitations such as a limited number of predefined tones and a limited number of frequency integrations in one tone. To work around these limitations and provide tone generation capability that is more flexible, the Prerecorded Tones (PRT) file can be used. If a specific prerecorded tone exists in the PRT file, it takes precedence over the same tone that exists in the CPT file and is played instead of it.

Note: The Prerecorded tones are used only for generation of tones. Detection of tones is performed according to the CPT file.

The PRT *dat* file contains a set of prerecorded tones to be played by the device during operation. Up to 40 tones (totaling approximately 10 minutes) can be stored in a single file in flash memory. The prerecorded tones are prepared offline using standard recording utilities (such as CoolEdit™) and combined into a single file using the TrunkPack Downloadable Conversion utility (refer to the *Product Reference Manual*).

The raw data files must be recorded with the following characteristics:

- **Coders:** G.711 A-law or G.711 µ-law
- **Rate:** 8 kHz
- **Resolution:** 8-bit
- **Channels:** mono

The generated PRT file can then be loaded to the device using the BootP/TFTP utility or via the Web interface (refer to "Loading Auxiliary Files" on page 266).

The prerecorded tones are played repeatedly. This enables you to record only part of the tone and play it for the full duration. For example, if a tone has a cadence of 2 seconds on and 4 seconds off, the recorded file should contain only these 6 seconds. The PRT module repeatedly plays this cadence for the configured duration. Similarly, a continuous tone can be played by repeating only part of it.

6.4 CAS Protocol Auxiliary Files

The CAS Protocol auxiliary files contain the CAS Protocol definitions that are used for CAS-terminated trunks. You can either use the supplied files or construct your own files. Up to eight files can be loaded and different files can be assigned to different trunks. The CAS files can be loaded to the device using the Web interface or *ini* file (refer to "Loading Auxiliary Files" on page 266).

Note: All CAS files loaded together must belong to the same Trunk Type (either E1 or T1).

6.5 Coefficient File

The Coeff_FXS.dat file is used to provide best termination and transmission quality adaptation for different line types for FXS interfaces. This adaptation is performed by modifying the telephony interface characteristics (such as DC and AC impedance, feeding current, and ringing voltage). The *coeff.dat* auxiliary file is produced specifically for each market after comprehensive performance analysis and testing, and can be modified on request. The current file supports US line type of 600 ohm AC impedance and 40 V RMS ringing voltage for REN = 2.

Note: The Coefficient File is applicable only to Mediant 1000.

To load the *coeff.dat* file to the device, use the Web interface or specify the FXS *coeff.dat* file name in the device's *ini* file (refer to "Loading Auxiliary Files" on page 266).

The *Coeff.dat* file consists of a set of parameters for the signal processor of the loop interface devices. This parameter set provides control of the following AC and DC interface parameters:

- DC (battery) feed characteristics
- AC impedance matching
- Transmit gain
- Receive gain
- Hybrid balance
- Frequency response in transmit and receive direction
- Hook thresholds
- Ringing generation and detection parameters

This means, for example, that changing impedance matching or hybrid balance doesn't require hardware modifications, so that a single device is able to meet requirements for different markets. The digital design of the filters and gain stages also ensures high reliability, no drifts (over temperature or time) and simple variations between different line types.

In future software releases, it is to be expanded to consist of different sets of line parameters, which can be selected in the *ini* file, for each port.

6.6 Dial Plan File

The source file for the Dial Plan configuration contains a list of known prefixes (e.g. area codes and international telephone number patterns) for the PSTN to which the device is connected. The device uses this information to detect end-of-dialing in certain CAS configurations where the end-indicator (ST) is not used. The device supports up to 8,000 distinct prefixes in the dial-plan file.

The CasTrunkDialPlanName *ini* file parameter determines which Dial Plan (in a Dial Plan file) to use for a specific trunk (refer to "Configuring the Trunk Settings" on page 96). The Dial Plan can be loaded using the Web interface (refer to "Loading Auxiliary Files" on page 266).

The following is an example of an *ini* file that includes these definitions. This *ini* file is converted (using the TrunkPack Conversion Utility - refer to the *Product Reference Manual*) to a binary file and loaded to the device.

```
; Example of dial-plan configuration.
; This file contains two dial plans: you may specify which
; one to use in CAS configuration.
[ PLAN1 ]
; Define the area codes 02, 03, 04.
; In these area codes, phone numbers have 7 digits.
02,7
03,7
04,7
; Define the cellular/VoIP area codes 052, 054, 050, and 077.
; In these area codes, phone numbers have 8 digits.
052,8
054,8
050,8
077,8
; Define the international prefixes 00, 012, 014.
; The number following these prefixes may
; be 7 to 14 digits in length.
00,7-14
012,7-14
014,7-14
; Define the emergency number 911.
; No additional digits are expected.
911,0
[ PLAN2 ]
; Define the area codes 02, 03, 04.
; In these area codes, phone numbers have 7 digits.
0[2-4],7
; Operator services starting with a star: *41, *42, *43.
; No additional digits are expected.
*4[1-3],0
```

The list must be prepared in a textual *ini* file with the following syntax:

- Every line in the file defines a known dialing prefix and the number of digits expected to follow that prefix. The prefix must be separated from the number of additional digits by a comma (',').
- Empty lines are ignored.
- Lines beginning with a semicolon (';') are ignored.
- Multiple dial plans may be specified in one file; A name in square brackets on a separate line indicates the beginning of a new dial plan. Up to eight dial plans can be defined.
- Asterisks ('*') and number-signs ('#') can be specified as part of the prefix.
- Numeric ranges are allowed in the prefix.
- A numeric range is allowed in the number of additional digits.

Note: The prefixes must not overlap. Attempting to process an overlapping configuration in the TrunkPack Conversion Utility results in an error message specifying the problematic line.

6.7 User Information File

The User Information file is a text file that maps PBX extensions, connected to the device, to global IP numbers. In this context, a global IP phone number (alphanumerical) serves as a routing identifier for calls in the 'IP World'. The PBX extension uses this mapping to emulate the behavior of an IP phone.

Note: The mapping mechanism is disabled by default and must be activated using the parameter EnableUserInfoUsage (refer to "Advanced Parameters" on page 166).

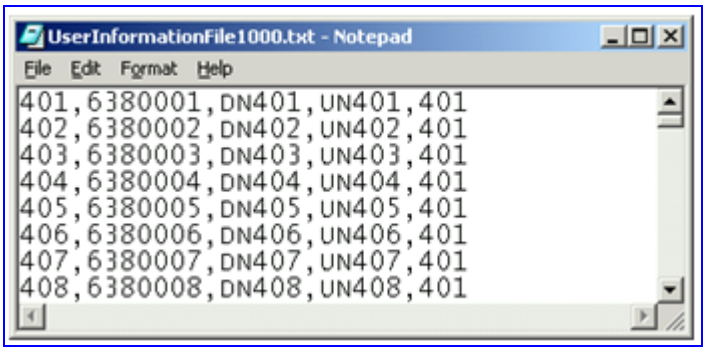
Each line in the file represents a mapping rule of a single PBX extension. Up to 1,00 rules can be configured. Each line includes five items separated with commas. The items are described in the table below:

Table 6-1: User Information Items

Item	Description	Maximum Size (Characters)
PBX extension #	The relevant PBX extension number.	10
Global phone #	The relevant global phone number.	20
Display name	A string that represents the PBX extensions for the Caller ID.	30
Username	A string that represents the user name for SIP registration.	40
Password	A string that represents the password for SIP registration.	20

An example of a User Information file is shown in the figure below:

Figure 6-1: Example of a User Information File



Note: The last line in the User Information file must end with a carriage return (i.e., by pressing the <Enter> key).

The User Information file can be loaded to the device using the *ini* file (UserInfoFileName parameter described in "Auxiliary / Configuration Files Parameters" on page 383), the Web interface (refer to "Loading Auxiliary Files" on page 266), or by using the automatic update mechanism (UserInfoFileURL, refer to the *Product Reference Manual*).

The maximum permissible size of the file is 10,800 bytes for Mediant 1000 analog modules and 108,000 bytes for digital modules.

Each PBX extension registers separately (a REGISTER message is sent for each entry only if AuthenticationMode is set to Per Endpoint) using the IP number in the From / To headers. The REGISTER messages are sent gradually. Initially, the device sends requests according to the maximum number of allowed SIP dialogs (configured by the parameter NumberOfActiveDialogs). After each received response, the subsequent request is sent. Therefore, no more than NumberOfActiveDialogs dialogs are active simultaneously. The user name and password are used for SIP Authentication when required.

The calling number of outgoing Tel-to-IP calls is first translated to an IP number and then (if defined), the manipulation rules are performed. The Display Name is used in the From header in addition to the IP number. The called number of incoming IP-to-Tel calls is translated to a PBX extension only after manipulation rules (if defined) are performed.

7 IP Telephony Capabilities

This section describes the device's IP telephony capabilities.

7.1 Configuring the DTMF Transport Types

You can control the way DTMF digits are transported over the IP network to the remote endpoint, by using one of the following modes:

- **Using INFO message according to Nortel IETF draft:** DTMF digits are carried to the remote side in INFO messages. To enable this mode, define the following:
  - RxDTMFOption = 0 (*ini* file); 'Declare RFC 2833 in SDP' field = 'No' (Web interface -- refer to "DTMF & Dialing Parameters" on page 160)
  - TxDTMFOption = 1 (*ini* file); '1st to 5th Tx DTMF Option' field = 'INFO (Nortel)' (Web interface -- refer to "DTMF & Dialing Parameters" on page 160)

  Note that in this mode, DTMF digits are erased from the audio stream [DTMFTransportType is automatically set to 0 ('DTMF Transport Type' field = 'DTMF Mute' -- Web interface)].
- **Using INFO message according to Cisco's mode:** DTMF digits are carried to the remote side in INFO messages. To enable this mode, define the following:
  - RxDTMFOption = 0 (*ini* file); 'Declare RFC 2833 in SDP' field = 'No' (Web interface -- refer to "DTMF & Dialing Parameters" on page 160)
  - TxDTMFOption = 3 (*ini* file); '1st to 5th Tx DTMF Option' field = 'INFO (Cisco)' (Web interface -- refer to "DTMF & Dialing Parameters" on page 160)

  Note that in this mode, DTMF digits are erased from the audio stream (DTMFTransportType is automatically set to 0 ('DTMF Transport Type' field = 'DTMF Mute' -- Web interface)].
- **Using NOTIFY messages according to <draft-mahy-sipping-signaled-digits-01.txt>**: DTMF digits are carried to the remote side using NOTIFY messages. To enable this mode, define the following:
  - RxDTMFOption = 0 (*ini* file); 'Declare RFC 2833 in SDP' field = 'No' (Web interface -- refer to "DTMF & Dialing Parameters" on page 160)
  - TxDTMFOption = 2 (*ini* file); '1st to 5th Tx DTMF Option' field = 'NOTIFY' (Web interface -- refer to "DTMF & Dialing Parameters" on page 160)

  Note that in this mode, DTMF digits are erased from the audio stream (DTMFTransportType is automatically set to 0 ('DTMF Transport Type' field = 'DTMF Mute' -- Web interface)].
- **Using RFC 2833 relay with Payload type negotiation:** DTMF digits are carried to the remote side as part of the RTP stream in accordance with RFC 2833 standard. To enable this mode, define the following:
  - RxDTMFOption = 3 (*ini* file); 'Declare RFC 2833 in SDP' field = 'Yes' (Web interface -- refer to "DTMF & Dialing Parameters" on page 160)
  - TxDTMFOption = 4 (*ini* file); '1st to 5th Tx DTMF Option' field = 'RFC 2833' (Web interface -- refer to "DTMF & Dialing Parameters" on page 160)

Note that to set the RFC 2833 payload type with a different value (other than its default, 96) configure the RFC2833PayloadType (RFC 2833 Payload Type) parameter. The device negotiates the RFC 2833 payload type using local and remote SDP and sends packets using the payload type from the received SDP. The device expects to receive RFC 2833 packets with the same payload type as configured by the RFC2833PayloadType parameter. If the remote side doesn't include 'telephony-event' in its SDP, the device sends DTMF digits in transparent mode (as part of the voice stream).

- **Sending DTMF digits (in RTP packets) as part of the audio stream (DTMF Relay is disabled):** This method is typically used with G.711 coders; with other low-bit rate (LBR) coders, the quality of the DTMF digits is reduced. To enable this mode, define the following:
  - RxDTMFOption = 0 (*ini* file); 'Declare RFC 2833 in SDP' field = 'No' (Web interface -- refer to "DTMF & Dialing Parameters" on page 160)
  - TxDTMFOption = 0 (*ini* file); '1st to 5th Tx DTMF Option' field = 'Disable' (Web interface -- refer to "DTMF & Dialing Parameters" on page 160)
  - DTMFTransportType = 2 (DTMF Transport Type = Transparent DTMF)
- **Using INFO message according to Korea mode:** DTMF digits are carried to the remote side in INFO messages. To enable this mode, define the following:
  - RxDTMFOption = 0 (*ini* file); 'Declare RFC 2833 in SDP' field = 'No' (Web interface -- refer to "DTMF & Dialing Parameters" on page 160)
  - TxDTMFOption = 3 (*ini* file); '1st to 5th Tx DTMF Option' field = 'INFO (Korea)' (Web interface -- refer to "DTMF & Dialing Parameters" on page 160)

  Note that in this mode, DTMF digits are erased from the audio stream (DTMFTransportType is automatically set to 0 (DTMF Mute)).

Notes:

- The device is always ready to receive DTMF packets over IP in all possible transport modes: INFO messages, NOTIFY, and RFC 2833 (in proper payload type) or as part of the audio stream.
- To exclude RFC 2833 Telephony event parameter from the device's SDP, set RxDTMFOption to 0 in the *ini* file.

The following parameters affect the way the device handles the DTMF digits:

- TxDTMFOption, RxDTMFOption, and RFC2833PayloadType (described in "DTMF & Dialing Parameters" on page 160)
- MGCPDTMFDetectionPoint, DTMFVolume, DTMFTransportType, DTMFDigitLength, and DTMFInterDigitInterval (refer to "Channel Parameters" on page 377)

7.2 Fax and Modem Capabilities

7.2.1 Fax/Modem Operating Modes

The device supports two modes of operations:

- Fax / modem negotiation that isn't performed during the establishment of the call.
- VBD mode for V.152 implementation (refer to "Supporting V.152 Implementation" on page 405): fax / modem capabilities are negotiated between the device and the

remote endpoint at the establishment of the call. During a call, when a fax / modem signal is detected, transition from voice to VBD (or T.38) is automatically performed and no additional SIP signaling is required. If negotiation fails (i.e., no match is achieved for any of the transport capabilities), fallback to existing logic occurs (according to the parameter IsFaxUsed).

7.2.2 Fax/Modem Transport Modes

The device supports the following transport modes for fax per modem type (V.22/V.23/Bell/V.32/V.34):

- T.38 fax relay (refer to "Fax Relay Mode" on page 399)
- Fax and modem bypass: a proprietary method that uses a high bit rate coder (refer to "Fax/Modem Bypass Mode" on page 400)
- NSE Cisco's Pass-through bypass mode for fax and modem (refer to "Fax / Modem NSE Mode" on page 401)
- Transparent: passing the fax / modem signal in the current voice coder (refer to "Fax / Modem Transparent Mode" on page 403)
- Transparent with events: passing the fax / modem signal in the current voice coder with adaptations (refer to "Fax / Modem Transparent with Events Mode" on page 404)
- G.711 Transport: switching to G.711 when fax/modem is detected (refer to "G.711 Fax / Modem Transport Mode" on page 402)
- Fax fallback to G.711 if T.38 is not supported (refer to "Fax Fallback" on page 402)

'Adaptations' refer to automatic reconfiguration of certain DSP features for handling fax/modem streams differently than voice.

7.2.2.1 T.38 Fax Relay Mode

In Fax Relay mode, fax signals are transferred using the T.38 protocol. T.38 is an ITU standard for sending fax across IP networks in real-time mode. The device currently supports only the T.38 UDP syntax.

T.38 can be configured in the following ways:

- Switching to T.38 mode using SIP Re-INVITE messages (refer to "Switching to T.38 Mode using SIP Re-INVITE" on page 400)
- Automatically switching to T.38 mode without using SIP Re-INVITE messages (refer to "Automatically Switching to T.38 Mode without SIP Re-INVITE" on page 400)

When fax transmission ends, the reverse switching from fax relay to voice is automatically performed at both the local and remote endpoints.

You can change the fax rate declared in the SDP, using the parameter FaxRelayMaxRate (this parameter doesn't affect the actual transmission rate). In addition, you can enable or disable Error Correction Mode (ECM) fax mode using the FaxRelayECMEnable parameter.

When using T.38 mode, you can define a redundancy feature to improve fax transmission over congested IP networks. This feature is activated using the FaxRelayRedundancyDepth and FaxRelayEnhancedRedundancyDepth parameters. Although this is a proprietary redundancy scheme, it should not create problems when working with other T.38 decoders.

7.2.2.1.1 Switching to T.38 Mode using SIP Re-INVITE

In the Switching to T.38 Mode using SIP Re-INVITE mode, upon detection of a fax signal, the terminating device negotiates T.38 capabilities using a Re-INVITE message. If the far-end device doesn't support T.38, the fax fails. In this mode, the parameter FaxTransportMode is ignored.

To configure T.38 mode using SIP Re-INVITE messages, set IsFaxUsed to 1. Additional configuration parameters include the following:

- FaxRelayEnhancedRedundancyDepth
- FaxRelayRedundancyDepth
- FaxRelayECMEnable
- FaxRelayMaxRate

7.2.2.1.2 Automatically Switching to T.38 Mode without SIP Re-INVITE

In the Automatically Switching to T.38 Mode without SIP Re-INVITE mode, when a fax signal is detected, the channel automatically switches from the current voice coder to answer tone mode, and then to T.38-compliant fax relay mode.

To configure automatic T.38 mode, perform the following configurations:

- IsFaxUsed = 0
- FaxTransportMode = 1
- Additional configuration parameters:
  - FaxRelayEnhancedRedundancyDepth
  - FaxRelayRedundancyDepth
  - FaxRelayECMEnable
  - FaxRelayMaxRate

7.2.2.2 Fax/Modem Bypass Mode

In this proprietary mode, when fax or modem signals are detected, the channel automatically switches from the current voice coder to a high bit-rate coder (according to the parameter FaxModemBypassCoderType). In addition, the channel is automatically reconfigured with the following fax / modem adaptations: disables silence suppression, enables echo cancellation for fax and disables it for modem, and performs certain jitter buffering optimizations. The network packets generated and received during the bypass period are regular voice RTP packets (per the selected bypass coder), but with a different RTP payload type (according to the parameters FaxBypassPayloadType and ModemBypassPayloadType).

During the bypass period, the coder uses the packing factor, which is defined by the parameter FaxModemBypassM. The packing factor determines the number of coder payloads (each the size of FaxModemBypassBasicRTPPacketInterval) that are used to generate a single fax/modem bypass packet.

When fax/modem transmission ends, the reverse switching, from bypass coder to regular voice coder is carried out.

To configure fax / modem bypass mode, perform the following configurations:

- IsFaxUsed = 0
- FaxTransportMode = 2
- V21ModemTransportType = 2
- V22ModemTransportType = 2
- V23ModemTransportType = 2
- V32ModemTransportType = 2
- V34ModemTransportType = 2
- BellModemTransportType = 2
- Additional configuration parameters:
  - FaxModemBypassCoderType
  - FaxBypassPayloadType
  - ModemBypassPayloadType
  - FaxModemBypassBasicRTPPacketInterval
  - FaxModemBypassDJBufMinDelay

Note: When the device is configured for modem bypass and T.38 fax, V.21 low-speed modems are not supported and fail as a result.

Tip: When the remote (non-AudioCodes') gateway uses G711 coder for voice and doesn't change the coder payload type for fax or modem transmission, it is recommended to use the Bypass mode with the following configuration:

- EnableFaxModemInbandNetworkDetection = 1
- FaxModemBypassCoderType = same coder used for voice
- FaxModemBypassM = same interval as voice
- ModemBypassPayloadType = 8 if voice coder is A-Law; 0 if voice coder is Mu-Law

7.2.2.3 Fax / Modem NSE Mode

In this mode, fax and modem signals are transferred using Cisco-compatible Pass-through bypass mode. Upon detection of fax or modem answering tone signal, the terminating device sends three to six special NSE RTP packets (using NSEpayloadType, usually 100). These packets signal the remote device to switch to G.711 coder (according to the parameter FaxModemBypassCoderType). After a few NSE packets are exchanged between the devices, both devices start using G.711 packets with standard payload type (8 for G.711 A-Law and 0 for G.711 Mu-Law). In this mode, no Re-INVITE messages are sent. The voice channel is optimized for fax/modem transmission (same as for usual bypass mode).

The parameters defining payload type for the proprietary AudioCodes' Bypass mode FaxBypassPayloadType and ModemBypassPayloadType are not used with NSE Bypass.

When configured for NSE mode, the device includes in its SDP the following line:

```
a=rtpmap:100 X-NSE/8000
```

(where 100 is the NSE payload type)

The Cisco gateway must include the following definition: "modem passthrough nse payload-type 100 codec g711alaw".

To configure NSE mode, perform the following configurations:

- IsFaxUsed = 0
- FaxTransportMode = 2
- NSEMode = 1
- NSEPayloadType = 100
- V21ModemTransportType = 2
- V22ModemTransportType = 2
- V23ModemTransportType = 2
- V32ModemTransportType = 2
- V34ModemTransportType = 2
- BellModemTransportType = 2

7.2.2.4 G.711 Fax / Modem Transport Mode

In this mode, when the terminating device detects fax or modem signals (CED or AnsAM), it sends a Re-INVITE message to the originating device requesting it to re-open the channel in G.711 VBD with the following adaptations:

- Echo Canceller = off
- Silence Compression = off
- Echo Canceller Non-Linear Processor Mode = off
- Dynamic Jitter Buffer Minimum Delay = 40
- Dynamic Jitter Buffer Optimization Factor = 13

After a few seconds upon detection of fax V.21 preamble or super G3 fax signals, the device sends a second Re-INVITE enabling the echo canceller (the echo canceller is disabled only on modem transmission).

A 'gpmd' attribute is added to the SDP according to the following format:

- **For G.711A-law:** a=gpmd:0 vbd=yes;ecan=on (or off, for modems)
- **For G.711 µ-law:** a=gpmd:8 vbd=yes;ecan=on (or off for modems)

The parameters FaxTransportMode and VxxModemTransportType are ignored and automatically set to the mode called 'transparent with events'.

To configure fax / modem transparent mode, set IsFaxUsed to 2.

7.2.2.5 Fax Fallback

In this mode, when the terminating device detects a fax signal, it sends a Re-INVITE message to the originating device with T.38. If the remote device doesn't support T.38 (replies with 415 Media Not Supported), the device sends a new Re-INVITE with G.711 VBD with the following adaptations:

- Echo Canceller = on
- Silence Compression = off
- Echo Canceller Non-Linear Processor Mode = off
- Dynamic Jitter Buffer Minimum Delay = 40
- Dynamic Jitter Buffer Optimization Factor = 13

When the device initiates a fax session using G.711, a 'gpmd' attribute is added to the SDP according to the following format:

- **For G.711A-law:** a=gpmd:0 vbd=yes;ecan=on
- **For G.711 µ-law:** a=gpmd:8 vbd=yes;ecan=on

In this mode, the parameter FaxTransportMode is ignored and automatically set to 'transparent'.

To configure fax fallback mode, set IsFaxUsed to 3.

7.2.2.6 Fax / Modem Transparent Mode

In this mode, fax and modem signals are transferred using the current voice coder without notifications to the user and without automatic adaptations. It's possible to use the Profiles mechanism (refer to "Configuring the Profile Definitions" on page 212) to apply certain adaptations to the channel used for fax / modem (e.g., to use the coder G.711, to set the jitter buffer optimization factor to 13, and to enable echo cancellation for fax and disable it for modem).

To configure fax / modem transparent mode, use the following parameters:

- IsFaxUsed = 0
- FaxTransportMode = 0
- V21ModemTransportType = 0
- V22ModemTransportType = 0
- V23ModemTransportType = 0
- V32ModemTransportType = 0
- V34ModemTransportType = 0
- BellModemTransportType = 0
- Additional configuration parameters:
  - CoderName
  - DJBufOptFactor
  - EnableSilenceCompression
  - EnableEchoCanceller

Note: This mode can be used for fax, but is not recommended for modem transmission. Instead, use the modes Bypass (refer to "Fax/Modem Bypass Mode" on page 400) or Transparent with Events (refer to "Fax / Modem Transparent with Events Mode" on page 404) for modem.

7.2.2.7 Fax / Modem Transparent with Events Mode

In this mode, fax and modem signals are transferred using the current voice coder with the following automatic adaptations:

- Echo Canceller = on (or off, for modems)
- Echo Canceller Non-Linear Processor Mode = off
- Jitter buffering optimizations

To configure fax / modem transparent with events mode, perform the following configurations:

- IsFaxUsed = 0
- FaxTransportMode = 3
- V21ModemTransportType = 3
- V22ModemTransportType = 3
- V23ModemTransportType = 3
- V32ModemTransportType = 3
- V34ModemTransportType = 3
- BellModemTransportType = 3

7.2.3 Supporting V.34 Faxes

Unlike T.30 fax machines, V.34 fax machines have no relay standard to transmit the data over IP to the remote side. Therefore, the device provides the following operation modes for transporting V.34 fax data over the IP:

- Using bypass mechanism for V.34 fax transmission (refer to "Using Bypass Mechanism for V.34 Fax Transmission" on page 404)
- Using relay mode, i.e., fallback to T.38 (refer to "Using Relay mode for both T.30 and V.34 faxes" on page 405)

Note: The CNG detector is disabled (CNGDetectorMode = 0) in all the following examples.

7.2.3.1 Using Bypass Mechanism for V.34 Fax Transmission

In this proprietary scenario, the device uses bypass (or NSE) mode to transmit V.34 faxes, enabling the full utilization of its speed.

Configure the following parameters to use bypass mode for both T.30 and V.34 faxes:

- FaxTransportMode = 2 (Bypass)
- V34ModemTransportType = 2 (Modem bypass)
- V32ModemTransportType = 2
- V23ModemTransportType = 2
- V22ModemTransportType = 2

Configure the following parameters to use bypass mode for V.34 faxes and T.38 for T.30 faxes:

- FaxTransportMode = 1 (Relay)
- V34ModemTransportType = 2 (Modem bypass)
- V32ModemTransportType = 2
- V23ModemTransportType = 2
- V22ModemTransportType = 2

7.2.3.2 Using Relay mode for both T.30 and V.34 faxes

In this scenario, V.34 fax machines are forced to use their backward compatibility with T.30 faxes and operate in the slower T.30 mode.

Use the following parameters to use T.38 mode for both V.34 faxes and T.30 faxes:

- FaxTransportMode = 1 (Relay)
- V34ModemTransportType = 0 (Transparent)
- V32ModemTransportType = 0
- V23ModemTransportType = 0
- V22ModemTransportType = 0

7.2.4 Supporting V.152 Implementation

The device supports the ITU-T recommendation V.152 (Procedures for Supporting Voice-Band Data over IP Networks). Voice-band data (VBD) is the transport of modem, facsimile, and text telephony signals over a voice channel of a packet network with a codec appropriate for such signals.

For V.152 capability, the device supports T.38 as well as VBD codecs (i.e., G.711 A-law and G.711 µ-law). The selection of capabilities is performed using the coders table (refer to "Coders" on page 157).

When in VBD mode for V.152 implementation, support is negotiated between the device and the remote endpoint at the establishment of the call. During this time, initial exchange of call capabilities is exchanged in the outgoing SDP. These capabilities include whether VBD is supported and associated RTP payload types ('gpmd' SDP attribute), supported codecs, and packetization periods for all codec payload types ('ptime' SDP attribute). After this initial negotiation, no Re-INVITE messages are necessary as both endpoints are synchronized in terms of the other side's capabilities. If negotiation fails (i.e., no match was achieved for any of the transport capabilities), fallback to existing logic occurs (according to the parameter IsFaxUsed).

Below is an example of media descriptions of an SDP indicating support for V.152.

```
v=0
o=-  0 0 IN IPV4 <IPAdressA>
s=-
t=0 0
p=+1
c=IN IP4  <IPAddressA
m=audio <udpPort A> RTP/AVP 18 0
a=ptime:10
a=rtpmap:96 PCMU/8000
a=gpmd: 96 vbd=yes
```

In the example, V.152 implementation is supported (using the dynamic payload type 96 and G.711 u-law as the VBD codec) as well as the voice codecs G.711 µ-law and G.729.

Instead of using VBD transport mode, the V.152 implementation can use alternative relay fax transport methods (e.g., fax relay over IP using T.38). The preferred V.152 transport method is indicated by the SDP 'pmft' attribute. Omission of this attribute in the SDP content means that VBD mode is the preferred transport mechanism for voice-band data.

To configure T.38 mode, use the CoderName parameter.

7.3 FXO Operating Modes

This section provides a description of the FXO operating modes and device configurations for Tel-to-IP and IP-to-Tel calls.

Note: This section is applicable only to Mediant 1000.

7.3.1 IP-to-Telephone Calls

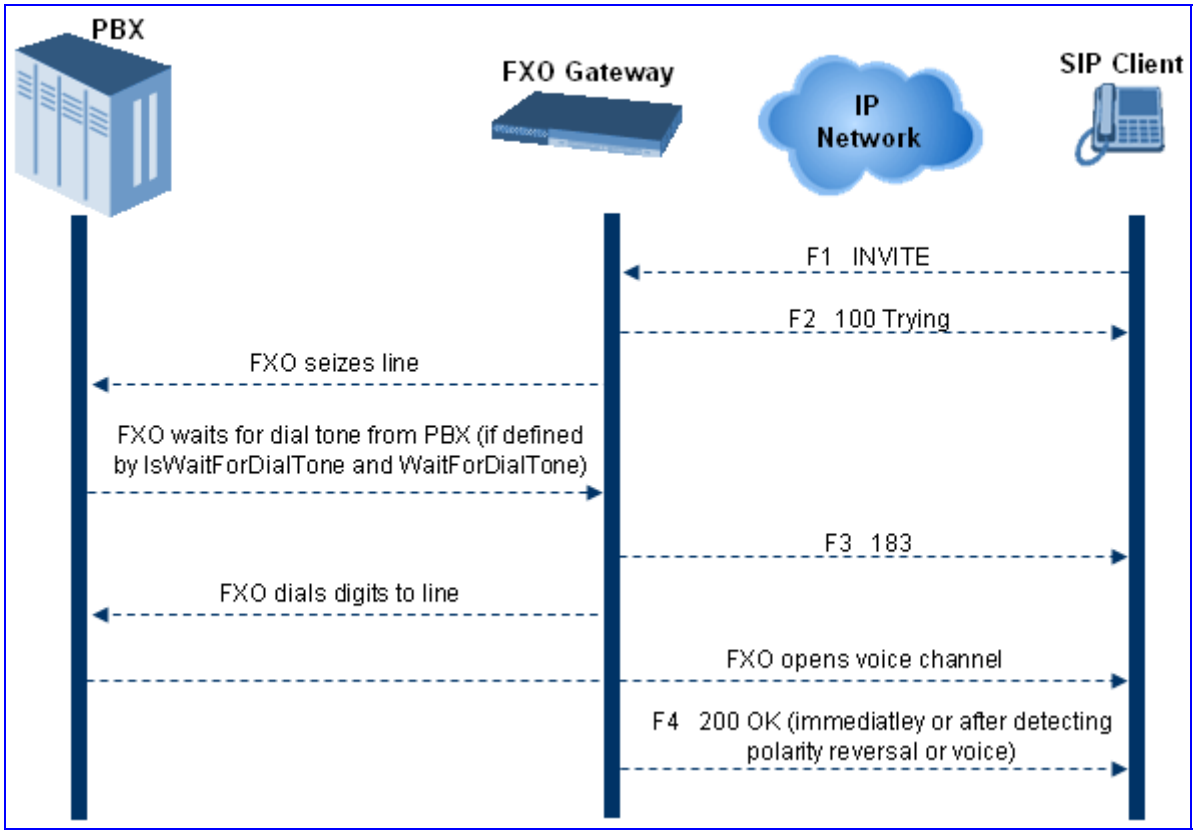
The FXO device provides the following operating modes for IP-to-Tel calls:

- One-stage dialing (refer to "One-Stage Dialing" on page 406)
  - Waiting for dial tone "Two-Stage Dialing" on page 408
  - Time to wait before dialing
  - Answer supervision
- Two-stage dialing (refer to "Two-Stage Dialing" on page 408)
- Dialing time
  - Disconnect supervision (refer to "Call Termination (Disconnect Supervision) on FXO Devices" on page 408)
  - DID wink (refer to "DID Wink" on page 409)

7.3.1.1 One-Stage Dialing

One-stage dialing is when the FXO device receives an IP-to-Tel call, off-hooks the PBX line connected to the telephone, and then immediately dials the destination telephone number. In other words, the IP caller doesn't dial the PSTN number upon hearing a dial tone.

Figure 7-1: Call Flow for One-Stage Dialing


PBX
FXO Gateway
IP Network
SIP Client
F1 INVITE
F2 100 Trying
FXO seizes line
FXO waits for dial tone from PBX (if defined by IsWaitForDialTone and WaitForDialTone)
F3 183
FXO dials digits to line
FXO opens voice channel
F4 200 OK (immediatley or after detecting polarity reversal or voice)

One-stage dialing incorporates the following FXO functionality:

- **Waiting for Dial Tone:** Enables the device to dial the digits to the Tel side only after detecting a dial tone from the PBX line. The *ini* file parameter IsWaitForDialTone is used to configure this operation.
- **Time to Wait Before Dialing:** Defines the time (in msec) between seizing the FXO line and starting to dial the digits. The *ini* file parameter WaitForDialTime is used to configure this operation.

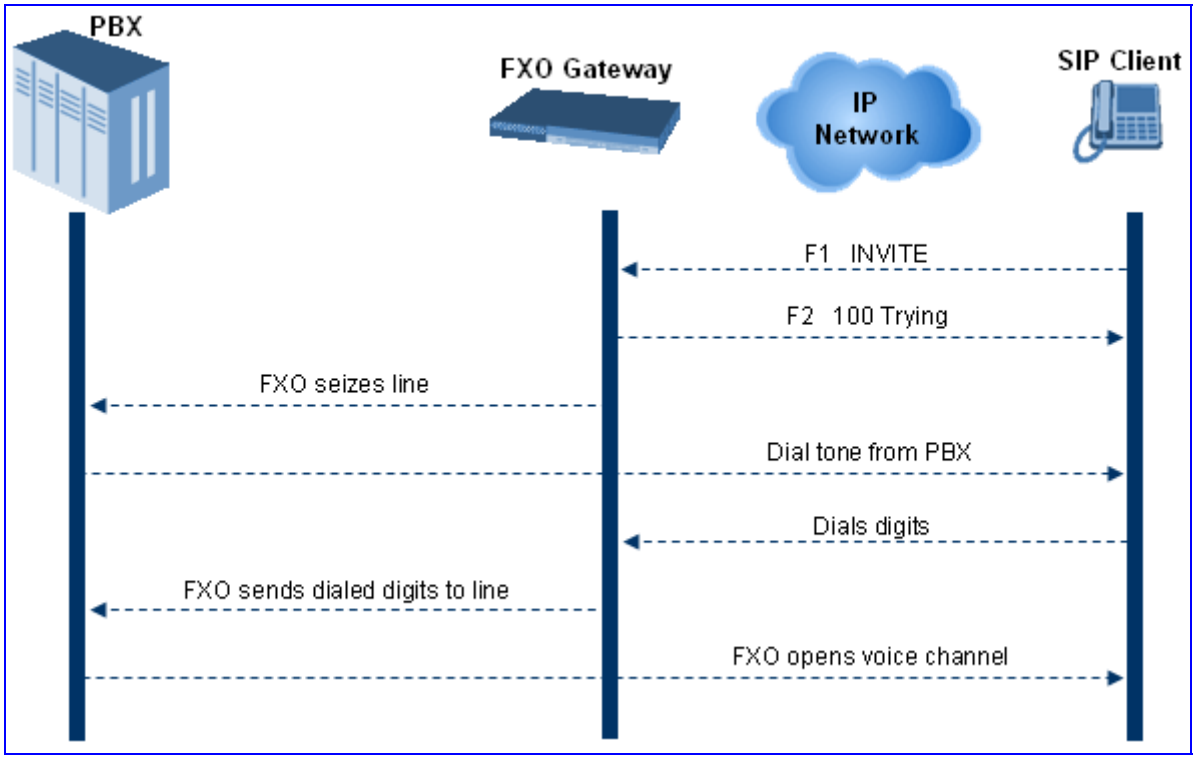
Note: The *ini* file parameter IsWaitForDialTone must be disabled for this mode.

- **Answer Supervision:** The Answer Supervision feature enables the FXO device to determine when a call is connected, by using one of the following methods:
  - Polarity Reversal: device sends a 200 OK in response to an INVITE only when it detects a polarity reversal.
  - Voice Detection: device sends a 200 OK in response to an INVITE only when it detects the start of speech (or ringback tone) from the Tel side. (Note that the IPM detectors must be enabled).

7.3.1.2 Two-Stage Dialing

Two-stage dialing is when the IP caller is required to dial twice. The caller initially dials to the FXO device and only after receiving a dial tone from the PBX (via the FXO device), dials the destination telephone number.

Figure 7-2: Call Flow for Two-Stage Dialing



Two-stage dialing implements the Dialing Time feature. Dialing Time allows you to define the time that each digit can be separately dialed. By default, the overall dialing time per digit is 200 msec. The longer the telephone number, the greater the dialing time.

The relevant parameters for configuring Dialing Time include the following:

- DTMFDigitLength (100 msec): time for generating DTMF tones to the PSTN (PBX) side
- DTMFInterDigitInterval (100 msec): time between generated DTMF digits to PSTN (PBX) side

7.3.1.3 Call Termination (Disconnect Supervision) on FXO Devices

The FXO Disconnect Supervision enables the device's FXO ports to monitor call-progress tones from a PBX or from the PSTN. This allows the FXO to determine when the call has terminated on the PBX side, and thereby, prevents analog trunks (i.e., lines to the PBX) from getting “stuck” when the called phone hangs up.

The PBX doesn't disconnect the call, but instead signals to the device that the call is disconnected using one of the following methods:

- **Detection of polarity reversal / current disconnect:** The call is immediately disconnected after polarity reversal or current disconnect is detected on the Tel side (assuming the PBX / CO produces this signal). This is the recommended method.

Relevant parameters: EnableReversalPolarity, EnableCurrentDisconnect, CurrentDisconnectDuration, CurrentDisconnectDefaultThreshold, and TimeToSampleAnalogLineVoltage.

- **Detection of Reorder, Busy, Dial, and Special Information Tone (SIT) tones:** The call is immediately disconnected after a Reorder, Busy, Dial, or SIT tone is detected on the Tel side (assuming the PBX / CO generates this tone). This method requires the correct tone frequencies and cadence to be defined in the Call Progress Tones file. If these frequencies are not known, define them in the CPT file (the tone produced by the PBX / CO must be recorded and its frequencies analyzed -- refer to Adding a Reorder Tone to the CPT File in the Reference Manual). This method is slightly less reliable than the previous one. You can use the CPTWizard (described in Call Progress Tones Wizard in the Reference Manual) to analyze Call Progress Tones generated by any PBX or telephone network.

  Relevant parameters: DisconnectOnBusyTone and DisconnectOnDialTone.

- **Detection of silence:** The call is disconnected after silence is detected on both call directions for a specific (configurable) amount of time. The call isn't disconnected immediately; therefore, this method should only be used as a backup option.

  Relevant parameters: EnableSilenceDisconnect and FarEndDisconnectSilencePeriod.

- **Special DTMF code:** A digit pattern that when received from the Tel side, indicates to the device to disconnect the call.

  Relevant *ini* file parameter: TelDisconnectCode.

- **Interruption of RTP stream:** Relevant parameters: BrokenConnectionEventTimeout and DisconnectOnBrokenConnection.

Note: This method operates correctly only if silence suppression is not used.

- **Protocol-based termination of the call from the IP side**

Note: The implemented disconnect method must be supported by the CO or PBX.

7.3.1.4 DID Wink

The device's FXO ports support Direct Inward Dialing (DID). DID is a service offered by telephone companies that enables callers to dial directly to an extension on a PBX without the assistance of an operator or automated call attendant. This service makes use of DID trunks, which forward only the last three to five digits of a phone number to the PBX. If, for example, a company has a PBX with extensions 555-1000 to 555-1999, and a caller dials 555-1234, the local central office (CO) would forward, for example, only 234 to the PBX. The PBX would then ring extension 234.

DID wink enables the originating end to seize the line by going off-hook. It waits for acknowledgement from the other end before sending digits. This serves as an integrity check that identifies a malfunctioning trunk and allows the network to send a re-order tone to the calling party.

The "start dial" signal is a wink from the PBX to the FXO device. The FXO then sends the last four to five DTMF digits of the called number. The PBX uses these digits to complete the routing directly to an internal station (telephone or equivalent)

- DID Wink can be used for connection to EIA/TIA-464B DID Loop Start lines
- Both FXO (detection) and FXS (generation) are supported

7.3.2 Telephone-to-IP Calls

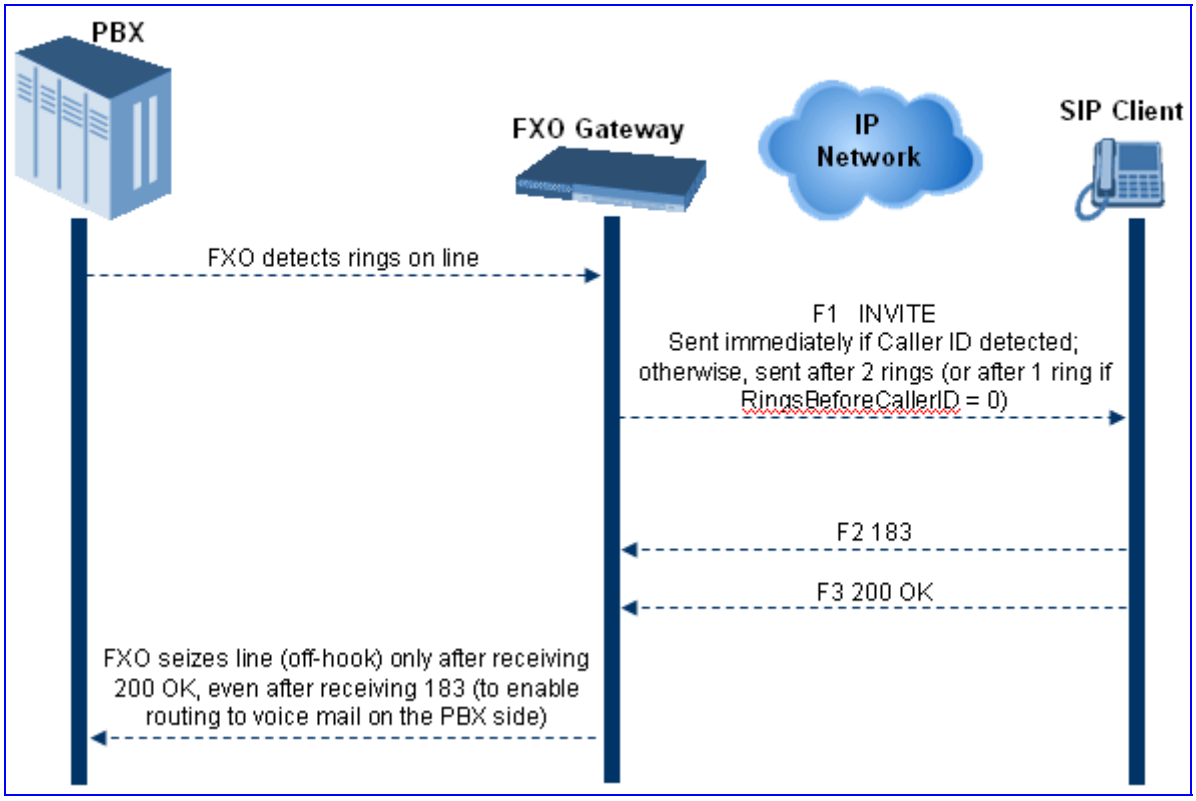
The FXO device provides the following FXO operating modes for Tel-to-IP calls:

- Automatic Dialing (refer to "Automatic Dialing" on page 410)
- Collecting Digits Mode (refer to "Collecting Digits Mode" on page 411)
- Ring Detection Timeout (refer to "Ring Detection Timeout" on page 411)
- FXO Supplementary Services (refer to "FXO Supplementary Services" on page 412)
  - Hold/Transfer Toward the Tel side
  - Hold/Transfer Toward the IP side
  - Blind Transfer to the Tel side

7.3.2.1 Automatic Dialing

Automatic dialing is defined using the *ini* file parameter table TargetOfChannel (refer to "Analog Telephony Parameters" on page 355) or the embedded Web server's 'Automatic Dialing' screen (refer to "Automatic Dialing" on page 220).

The SIP call flow diagram below illustrates Automatic Dialing.

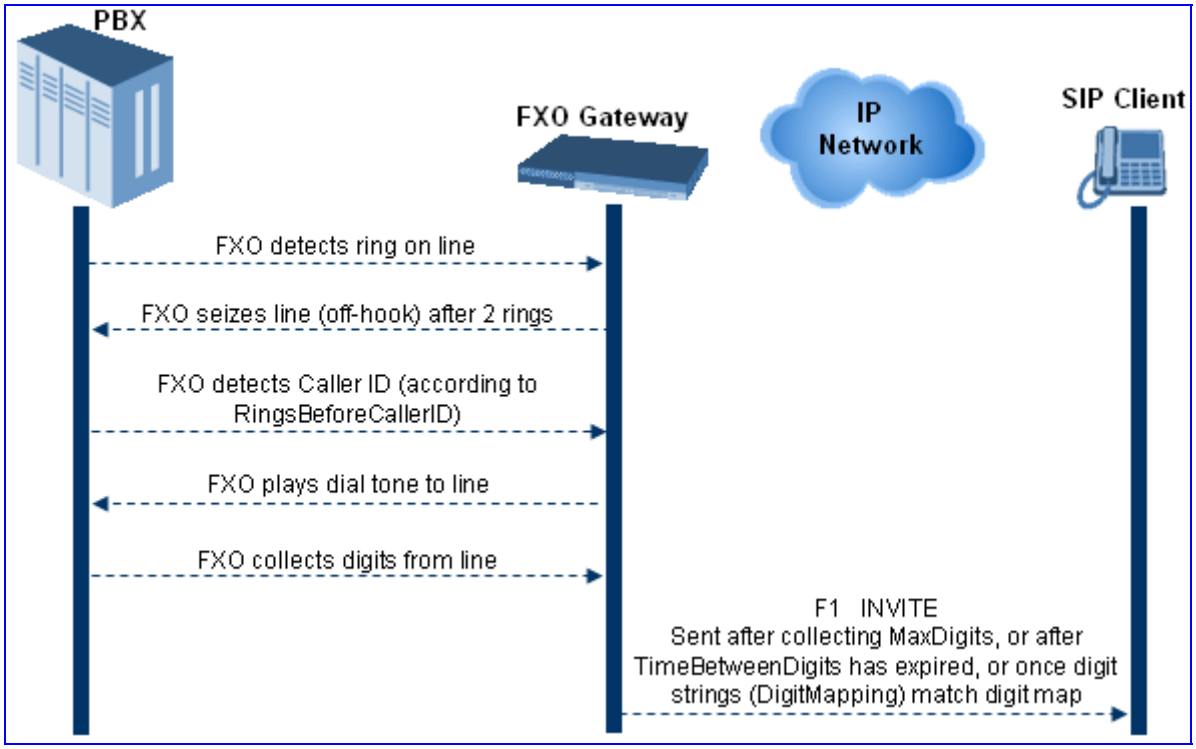

PBX
FXO Gateway
IP Network
SIP Client
FXO detects rings on line
F1 INVITE
Sent immediately if Caller ID detected; otherwise, sent after 2 rings (or after 1 ring if RingsBeforeCallerID = 0)
F2 183
F3 200 OK
FXO seizes line (off-hook) only after receiving 200 OK, even after receiving 183 (to enable routing to voice mail on the PBX side)

7.3.2.2 Collecting Digits Mode

When automatic dialing is not defined, the device collects the digits.

The SIP call flow diagram below illustrates the Collecting Digits Mode.

Figure 7-3: Call Flow for Collecting Digits Mode


PBX
FXO Gateway
IP Network
SIP Client
FXO detects ring on line
FXO seizes line (off-hook) after 2 rings
FXO detects Caller ID (according to RingsBeforeCallerID)
FXO plays dial tone to line
FXO collects digits from line
F1 INVITE
Sent after collecting MaxDigits, or after TimeBetweenDigits has expired, or once digit strings (DigitMapping) match digit map

7.3.2.3 Ring Detection Timeout

The *ini* file parameters IsWaitForDialTone and WaitForDialTone apply to Ring Detection Timeout. The operation of Ring Detection Timeout depends on the following:

- No automatic dialing and Caller ID is enabled: if the second ring signal doesn't arrive for Ring Detection Timeout, the device doesn't initiate a call to the IP.
- Automatic dialing is enabled: if the remote party doesn't answer the call, and the ringing signal stops for Ring Detection Timeout, the FXO releases the IP call.

Ring Detection Timeout supports full ring cycle of ring on and ring off (from ring start to ring start).

7.3.2.4 FXO Supplementary Services

The FXO supplementary services include the following:

- **Hold / Transfer toward the Tel side:** The *ini* file parameter LineTransferMode must be set to 0 (default). If the FXO receives a hook-flash from the IP side (using out-of-band or RFC 2833), the device sends the hook-flash to the Tel side by performing one of the following:
  - Performing a hook flash (i.e., on-hook and off-hook)
  - Sending a hook-flash code (defined by the *ini* file parameter HookFlashCode)

  The PBX may generate a dial tone that is sent to the IP, and the IP side may dial digits of a new destination.
- **Blind Transfer to the Tel side:** A blind transfer is one in which the transferring phone connects the caller to a destination line before ringback begins. The *ini* file parameter LineTransferMode must be set to 1.

  The blind transfer call process is as follows:
  - FXO receives a REFER request from the IP side
  - FXO sends a hook-flash to the PBX, dials the digits (that are received in the Refer-To header), and then drops the line (on-hook). Note that the time between flash to dial is according to the WaitForDialTime parameter.
  - PBX performs the transfer internally
- **Hold / Transfer toward the IP side:** The FXO device doesn't initiate hold / transfer as a response to input from the Tel side. If the FXO receives a REFER request (with or without replaces), it generates a new INVITE according to the Refer-To header.

7.4 Event Notification using X-Detect Header

The device supports the sending of notifications to a remote party notifying the occurrence (or detection) of certain events on the media stream. Event detection and notifications is performed using the X-Detect SIP message header, and only when establishing a SIP dialog.

For supporting some events, certain device configurations need to be performed. The table below lists the support event types (and subtypes) and the corresponding device configurations, if required:

Table 7-1: Supported X-Detect Event Types

Events		Required Configuration
Type	**Subtype**	
CPT	SIT	SITDetectorEnable = 1 UserDefinedToneDetectorEnable = 1
FAX	CED	(IsFaxUsed ≠ 0) or (IsFaxUsed = 0, and FaxTransportMode ≠ 0)
	modem	VxxModemTransportType = 3
PTT	voice-start voice-end	EnableDSPIPMDetectors = 1

The X-Detect event notification process is as follows:

1. For IP-to-Tel or Tel-to-IP calls, the device receives a SIP request message (using the X-Detect header) that the remote party wishes to detect events on the media stream. For incoming (IP-to-Tel) calls, the request must be indicated in the initial INVITE and responded to either in the 183 response (for early dialogs) or in the 200 OK response (for confirmed dialogs). For outgoing calls (Tel-to-IP), the request may be received in the 183 (for early dialogs) and responded to in the PRACK, or received in the 200 OK (for confirmed dialogs) and responded to in the ACK.
2. Once the device receives such a request, it sends a SIP response message (using the X-Detect header) to the remote party, listing all supported events that can be detected. The absence of the X-Detect header indicates that no detections are available.
3. Each time the device detects a supported event, the event is notified to the remote party, by sending an INFO message with the following message body:
   - Content-Type: application/X-DETECT
   - Type = [CPT | FAX | PTT…]
   - Subtype = xxx (according to the defined subtypes of each type)

Below is an example of SIP messages implementing the X-Detect header:

```
INVITE sip:101@10.33.2.53;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
Max-Forwards: 70
From: "anonymous" <sip:anonymous@anonymous.invalid>;tag=1c25298
To: <sip:101@10.33.2.53;user=phone>
Call-ID: 11923@10.33.2.53
CSeq: 1 INVITE
Contact: <sip:100@10.33.2.53>
X- Detect: Request=CPT,FAX
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
From: "anonymous" <sip:anonymous@anonymous.invalid>;tag=1c25298
To: <sip:101@10.33.2.53;user=phone>;tag=1c19282
Call-ID: 11923@10.33.2.53
CSeq: 1 INVITE
Contact: <sip:101@10.33.2.53>
X- Detect: Response=CPT,FAX
INFO sip:101@10.33.2.53;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
Max-Forwards: 70
From: "anonymous" <sip:anonymous@anonymous.invalid>;tag=1c25298
To: <sip:101@10.33.2.53;user=phone>
Call-ID: 11923@10.33.2.53
CSeq: 1 INVITE
Contact: <sip:100@10.33.2.53>
X- Detect: Response=CPT,FAX
Content-Type: Application/X-Detect
Content-Length: xxx
Type = CPT
Subtype = SIT
```

7.5 RTP Multiplexing (ThroughPacket)

The device supports a proprietary method to aggregate RTP streams from several channels to reduce the bandwidth overhead caused by the attached Ethernet, IP, UDP, and RTP headers, and to reduce the packet / data transmission rate. This option reduces the load on network routers and can typically save 50% (e.g., for G.723) on IP bandwidth. RTP Multiplexing (ThroughPacket™) is accomplished by aggregating payloads from several channels that are sent to the same destination IP address into a single IP packet.

RTP multiplexing can be applied to the entire device (refer to "Configuring the RTP / RTCP Settings" on page 88) or to specific IP destinations using the IP Profile feature (refer to "IP Profile Settings" on page 216).

To enable RTP Multiplexing, set the parameter RemoteBaseUDPPort to a nonzero value. Note that the value of RemoteBaseUDPPort on the local device must equal the value of BaseUDPPort of the remote device. The device uses these parameters to identify and distribute the payloads from the received multiplexed IP packet to the relevant channels.

In RTP Multiplexing mode, the device uses a single UDP port for all incoming multiplexed packets and a different port for outgoing packets. These ports are configured using the parameters L1L1ComplexTxUDPPort and L1L1ComplexRxUDPPort.

When RTP Multiplexing is used, call statistics aren't available (since there is no RTCP flow).

Notes:

- RTP Multiplexing must be enabled on both devices.
- When VLANs are imlemented, the RTP Multiplexing mechanism is not supported.

7.6 Dynamic Jitter Buffer Operation

Voice frames are transmitted at a fixed rate. If the frames arrive at the other end at the same rate, voice quality is perceived as good. In many cases, however, some frames can arrive slightly faster or slower than the other frames. This is called jitter (delay variation), and degrades the perceived voice quality. To minimize this problem, the device uses a jitter buffer. The jitter buffer collects voice packets, stores them and sends them to the voice processor in evenly spaced intervals.

The device uses a dynamic jitter buffer that can be configured using the following two parameters:

- **Minimum delay:** DJBufMinDelay (0 msec to 150 msec)
  Defines the starting jitter capacity of the buffer. For example, at 0 msec, there is no buffering at the start. At the default level of 10 msec, the device always buffers incoming packets by at least 10 msec worth of voice frames.

- **Optimization Factor:** DJBufOptFactor (0 to 12, 13)
  Defines how the jitter buffer tracks to changing network conditions. When set at its maximum value of 12, the dynamic buffer aggressively tracks changes in delay (based on packet loss statistics) to increase the size of the buffer and doesn't decay back down. This results in the best packet error performance, but at the cost of extra delay. At the minimum value of 0, the buffer tracks delays only to compensate for clock drift and quickly decays back to the minimum level. This optimizes the delay performance but at the expense of a higher error rate.

The default settings of 10 msec Minimum delay and 10 Optimization Factor should provide a good compromise between delay and error rate. The jitter buffer 'holds' incoming packets for 10 msec before making them available for decoding into voice. The coder polls frames from the buffer at regular intervals in order to produce continuous speech. As long as delays in the network do not change (jitter) by more than 10 msec from one packet to the next, there is always a sample in the buffer for the coder to use. If there is more than 10 msec of delay at any time during the call, the packet arrives too late. The coder tries to access a frame and is not able to find one. The coder must produce a voice sample even if a frame is not available. It therefore compensates for the missing packet by adding a Bad-Frame-Interpolation (BFI) packet. This loss is then flagged as the buffer being too small. The dynamic algorithm then causes the size of the buffer to increase for the next voice session. The size of the buffer may decrease again if the device notices that the buffer is not filling up as much as expected. At no time does the buffer decrease to less than the minimum size configured by the Minimum delay parameter.

For certain scenarios, the **Optimization Factor is set to 13**: One of the purposes of the Jitter Buffer mechanism is to compensate for clock drift. If the two sides of the VoIP call are not synchronized to the same clock source, one RTP source generates packets at a lower rate, causing under-runs at the remote Jitter Buffer. In normal operation (optimization factor 0 to 12), the Jitter Buffer mechanism detects and compensates for the clock drift by occasionally dropping a voice packet or by adding a BFI packet.

Fax and modem devices are sensitive to small packet losses or to added BFI packets. Therefore, to achieve better performance during modem and fax calls, the Optimization Factor should be set to 13. In this special mode the clock drift correction is performed less frequently - only when the Jitter Buffer is completely empty or completely full. When such condition occurs, the correction is performed by dropping several voice packets simultaneously or by adding several BFI packets simultaneously, so that the Jitter Buffer returns to its normal condition.

7.7 Configuring Alternative Routing (Based on Connectivity and QoS)

The Alternative Routing feature enables reliable routing of Tel-to-IP calls when a Proxy isn't used. The device periodically checks the availability of connectivity and suitable Quality of Service (QoS) before routing. If the expected quality cannot be achieved, an alternative IP route for the prefix (phone number) is selected.

Note: If the alternative routing destination is the device itself, the call can be configured to be routed back to one of the device's trunk groups and thus, back into the PSTN (PSTN Fallback).

7.7.1 Alternative Routing Mechanism

When a Tel-to-IP call is routed through the device, the call's destination number is compared to the list of prefixes defined in the 'Tel to IP Routing' table (described in "Tel to IP Routing Table" on page 200). The 'Tel to IP Routing' table is scanned for the destination number's prefix starting at the top of the table. For this reason, enter the main IP route above any alternative route. When an appropriate entry (destination number matches one of the prefixes) is found, the prefix's corresponding destination IP address is verified. If the destination IP address is disallowed (or if the original call fails and the device has made two additional attempts to establish the call without success), an alternative route is searched in the table. , after which an alternative route is used.

Destination IP address is disallowed if no ping to the destination is available (ping is continuously initiated every seven seconds), when an inappropriate level of QoS was detected, or when a DNS host name is not resolved. The QoS level is calculated according to delay or packet loss of previously ended calls. If no call statistics are received for two minutes, the QoS information is reset.

7.7.2 Determining the Availability of Destination IP Addresses

To determine the availability of each destination IP address (or host name) in the routing table, one (or all) of the following (configurable) methods are applied:

- **Connectivity:** The destination IP address is queried periodically (currently only by ping).
- **QoS:** The QoS of an IP connection is determined according to RTCP statistics of previous calls. Network delay (in msec) and network packet loss (in percentage) are separately quantified and compared to a certain (configurable) threshold. If the calculated amounts (of delay or packet loss) exceed these thresholds, the IP connection is disallowed.
- **DNS resolution:** When host name is used (instead of IP address) for the destination route, it is resolved to an IP address by a DNS server. Connectivity and QoS are then applied to the resolved IP address.

7.7.3 PSTN Fallback as a Special Case of Alternative Routing

The PSTN Fallback feature enables the device to redirect PSTN originated calls back to the legacy PSTN network if a destination IP route is unsuitable (disallowed) for voice traffic at a specific time. To enable PSTN fallback, assign the device's IP address as an alternative route to the desired prefixes. Note that calls (now referred to as IP-to-Tel calls) can be re-routed to a specific trunk group using the Routing parameters (refer to "IP to Trunk Group Routing" on page 204).

7.7.4 Relevant Parameters

The following parameters (described in "Routing General Parameters" on page 198) are used to configure the Alternative Routing mechanism:

- AltRoutingTel2IPEnable
- AltRoutingTel2IPMode
- IPConnQoSMaxAllowedPL
- IPConnQoSMaxAllowedDelay

7.8 Mapping PSTN Release Cause to SIP Response

The device's FXO interface interoperates between the SIP network and the PSTN/PBX. This interoperability includes the mapping of PSTN/PBX Call Progress Tones to SIP 4xx or 5xx responses for IP-to-Tel calls. The converse is also true: for Tel-to-IP calls, the SIP 4xx or 5xx responses are mapped to tones played to the PSTN/PBX.

When establishing an IP-to-Tel call, the following rules are applied:

- If the remote party (PSTN/PBX) is busy and the FXO device detects a Busy tone, it sends 486 Busy to IP. If it detects a Reorder tone, it sends 404 Not Found (no route to destination) to IP. In both cases the call is released. Note that if

DisconnectOnBusyTone is set to 0, the FXO device ignores the detection of Busy/Reorder tones and doesn't release the call.

- For all other FXS/FXO release types (caused when there are no free channels in the specific Trunk Group, or when an appropriate rule for routing the call to a Trunk Group doesn't exist, or if the phone number isn't found), the device sends a SIP response (to IP) according to the parameter DefaultReleaseCause. This parameter defines Q.931 release causes. Its default value is '3', which is mapped to the SIP 404 response. By changing its value to '34', the SIP 503 response is sent. Other causes can be used as well.

Note: This section is applicable only to Mediant 1000.

7.9 Call Detail Record

The Call Detail Record (CDR) contains vital statistic information on calls made by the device. CDRs are generated at the end and (optionally) at the beginning of each call (determined by the parameter CDRReportLevel), and then sent to a Syslog server. The destination IP address for CDR logs is determined by the parameter CDRSyslogServerIP. For CDR in RADIUS format, refer to "Supported RADIUS Attributes" on page 419.

The following table lists the CDR fields that are supported.

Table 7-2: Supported CDR Fields

Field Name	Description
ReportType	Report for either Call Started, Call Connected, or Call Released
Cid	Port Number
CallId	SIP Call Identifier
Trunk	Physical Trunk Number
BChan	Selected B-Channel
ConId	SIP Conference ID
TG	Trunk Group Number
EPTyp	Endpoint Type
Orig	Call Originator (IP, Tel)
SourceIp	Source IP Address
DestIp	Destination IP Address
TON	Source Phone Number Type
NPI	Source Phone Number Plan
SrcPhoneNum	Source Phone Number
SrcNumBeforeMap	Source Number Before Manipulation
TON	Destination Phone Number Type
NPI	Destination Phone Number Plan

Field Name	Description
DstPhoneNum	Destination Phone Number
DstNumBeforeMap	Destination Number Before Manipulation
Durat	Call Duration
Coder	Selected Coder
Intrv	Packet Interval
RtpIp	RTP IP Address
Port	Remote RTP Port
TrmSd	Initiator of Call Release (IP, Tel, Unknown)
TrmReason	Termination Reason
Fax	Fax Transaction during the Call
InPackets	Number of Incoming Packets
OutPackets	Number of Outgoing Packets
PackLoss	Local Packet Loss
RemotePackLoss	Number of Outgoing Lost Packets
UniqueId	unique RTP ID
SetupTime	Call Setup Time
ConnectTime	Call Connect Time
ReleaseTime	Call Release Time
RTPdelay	RTP Delay
RTPjitter	RTP Jitter
RTPssrc	Local RTP SSRC
RemoteRTPssrc	Remote RTP SSRC
RedirectReason	Redirect Reason
TON	Redirection Phone Number Type
MeteringPulses	Number of Generated Metering Pulses
NPI	Redirection Phone Number Plan
RedirectPhonNum	Redirection Phone Number

7.10 Supported RADIUS Attributes

Use the following table for explanations on the RADIUS attributes contained in the communication packets transmitted between the device and a RADIUS Server.

Table 7-3: Supported RADIUS Attributes

Attribute Number	Attribute Name	VSA No.	Purpose	Value Format	Example	AAA[1]
Request Attributes						
1	User-Name		Account number or calling party number or blank	String up to 15 digits long	5421385747	Start Acc Stop Acc
4	NAS-IP-Address		IP address of the requesting device	Numeric	192.168.14.43	Start Acc Stop Acc
6	Service-Type		Type of service requested	Numeric	1: login	Start Acc Stop Acc
26	H323-Incoming-Conf-Id	1	SIP call identifier	Up to 32 octets		Start Acc Stop Acc
26	H323-Remote-Address	23	IP address of the remote gateway	Numeric		Stop Acc
26	H323-Conf-ID	24	H.323/SIP call identifier	Up to 32 octets		Start Acc Stop Acc
26	H323-Setup-Time	25	Setup time in NTP format 1	String		Start Acc Stop Acc
26	H323-Call-Origin	26	The call's originator: Answering (IP) or Originator (PSTN)	String	Answer, Originate etc	Start Acc Stop Acc
26	H323-Call-Type	27	Protocol type or family used on this leg of the call	String	VoIP	Start Acc Stop Acc
26	H323-Connect-Time	28	Connect time in NTP format	String		Stop Acc
26	H323-Disconnect-Time	29	Disconnect time in NTP format	String		Stop Acc
26	H323-Disconnect-Cause	30	Q.931 disconnect cause code	Numeric		Stop Acc
26	H323-Gw-ID	33	Name of the gateway	String	SIPIDString	Start Acc Stop Acc
26	SIP-Call-ID	34	SIP Call ID	String	abcde@ac.com	Start Acc Stop Acc
26	Call-Terminator	35	The call's terminator: PSTN-terminated call (Yes); IP-terminated call (No).	String	Yes, No	Stop Acc
30	Called-Station-			String	8004567145	Start Acc

Attribute Number	Attribute Name	VSA No.	Purpose	Value Format	Example	AAA[1]
	ID		Destination phone number	String	2427456425	Stop Acc
31	Calling-Station-ID		Calling Party Number (ANI)	String	5135672127	Start Acc Stop Acc
40	Acct-Status-Type		Account Request Type (start or stop) **Note:** 'start' isn't supported on the Calling Card application.	Numeric	1: start, 2: stop	Start Acc Stop Acc
41	Acct-Delay-Time		No. of seconds tried in sending a particular record	Numeric	5	Start Acc Stop Acc
42	Acct-Input-Octets		Number of octets received for that call duration	Numeric		Stop Acc
43	Acct-Output-Octets		Number of octets sent for that call duration	Numeric		Stop Acc
44	Acct-Session-ID		A unique accounting identifier - match start & stop	String	34832	Start Acc Stop Acc
46	Acct-Session-Time		For how many seconds the user received the service	Numeric		Stop Acc
47	Acct-Input-Packets		Number of packets received during the call	Numeric		Stop Acc
48	Acct-Output-Packets		Number of packets sent during the call	Numeric		Stop Acc
61	NAS-Port-Type		Physical port type of device on which the call is active	String	0: Asynchronous	Start Acc Stop Acc
Response Attributes						
26	H323-Return-Code	103	The reason for failing authentication (0 = ok, other number failed)	Numeric	0 Request accepted	Stop Acc
44	Acct-Session-ID		A unique accounting identifier – match start & stop	String		Stop Acc

Below is an example of RADIUS Accounting, where the non-standard parameters are preceded with brackets.

```
Accounting-Request (361)
user-name = 111
acct-session-id = 1
nas-ip-address = 212.179.22.213
nas-port-type = 0
acct-status-type = 2
acct-input-octets = 4841
acct-output-octets = 8800
acct-session-time = 1
acct-input-packets = 122
acct-output-packets = 220
called-station-id = 201
calling-station-id = 202
// Accounting non-standard parameters:
(4923 33) h323-gw-id =
(4923 23) h323-remote-address = 212.179.22.214
(4923 1) h323-ivr-out = h323-incoming-conf-id:02102944 600a1899
3fd61009 0e2f3cc5
(4923 30) h323-disconnect-cause = 22 (0x16)
(4923 27) h323-call-type = VOIP
(4923 26) h323-call-origin = Originate
(4923 24) h323-conf-id = 02102944 600a1899 3fd61009 0e2f3cc5
```

7.11 Trunk-to-Trunk Routing Example

This example describes two devices, each interfacing with the PSTN through four E1 spans. Device **A** is configured to route all incoming Tel-to-IP calls to Device **B**. Device **B** generates calls to the PSTN on the same E1 trunk on which the call was originally received (in Device **A**).

- Device **A** IP address: 192.168.3.50
- Device **B** IP address: 192.168.3.51

The *ini* file parameters configuration for devices **A** and **B** include the following:

1. At both devices, define four trunk groups, each with 30 B-channels:
   - TrunkGroup_1 = 0/1-31,1000
   - TrunkGroup_2 = 1/1-31,2000
   - TrunkGroup_3 = 2/1-31,3000
   - TrunkGroup_4 = 3/1-31,4000
2. At Device **A**, add the originating Trunk Group ID as a prefix to the destination number for Tel-to-IP calls:

   AddTrunkGroupAsPrefix = 1
3. At Device **A**, route all incoming PSTN calls starting with prefixes 1, 2, 3, and 4, to the IP address of Device **B**:
   - Prefix = 1, 192.168.3.51
   - Prefix = 2, 192.168.3.51
   - Prefix = 3, 192.168.3.51
   - Prefix = 4, 192.168.3.51

   Note: You can also define Prefix = *,192.168.3.51, instead of the four lines above.

4. At Device **B**, route IP-to-PSTN calls to Trunk Group ID according to the first digit of the called number:
   - PSTNPrefix = 1,1
   - PSTNPrefix = 2,2
   - PSTNPrefix = 3,4
   - PSTNPrefix = 4,4
5. At Device **B**, remove the first digit from each IP-to-PSTN number before it is used in an outgoing call:

   NumberMapIP2Tel = *,1

7.12 Proxy or Registrar Registration Example

Below is an example of Proxy and Registrar Registration:

```
REGISTER sip:servername SIP/2.0
VIA: SIP/2.0/UDP 212.179.22.229;branch=z9hG4bRaC7AU234
From: <sip:GWRegistrationName@sipgatewayname>;tag=1c29347
To: <sip:GWRegistrationName@sipgatewayname>
Call-ID: 10453@212.179.22.229
Seq: 1 REGISTER
Expires: 3600
Contact: sip:GWRegistrationName@212.179.22.229
Content-Length: 0
```

The ‘servername’ string is defined according to the following rules:

- The "servername" is equal to "RegistrarName" if configured. The "RegistrarName" can be any string.
- Otherwise, the "servername" is equal to "RegistrarIP" (either FQDN or numerical IP address), if configured.
- Otherwise, the "servername" is equal to "ProxyName" if configured. The "ProxyName" can be any string.
- Otherwise, the "servername" is equal to "ProxyIP" (either FQDN or numerical IP address).

The parameter GWRegistrationName can be any string. This parameter is used only if registration is per device. If the parameter is not defined, the parameter UserName is used instead. If the registration is per endpoint, the endpoint phone number is used.

The 'sipgatewayname' parameter (defined in the *ini* file or Web interface) can be any string. Some Proxy servers require that the 'sipgatewayname' (in REGISTER messages) is set equal to the Registrar / Proxy IP address or to the Registrar / Proxy domain name. The 'sipgatewayname' parameter can be overwritten by the TrunkGroupSettings_GatewayName value if the TrunkGroupSettings_RegistrationMode is set to 'Per Endpoint'.

REGISTER messages are sent to the Registrar's IP address (if configured) or to the Proxy's IP address. A single message is sent once per device, or messages are sent per B-channel according to the parameter AuthenticationMode. There is also an option to configure registration mode per Trunk Group using the TrunkGroupSettings table. The registration request is resent according to the parameter RegistrationTimeDivider. For example, if RegistrationTimeDivider = 70 (%) and Registration Expires time = 3600, the device resends its registration request after 3600 x 70% = 2520 sec. The default value of RegistrationTimeDivider is 50%.

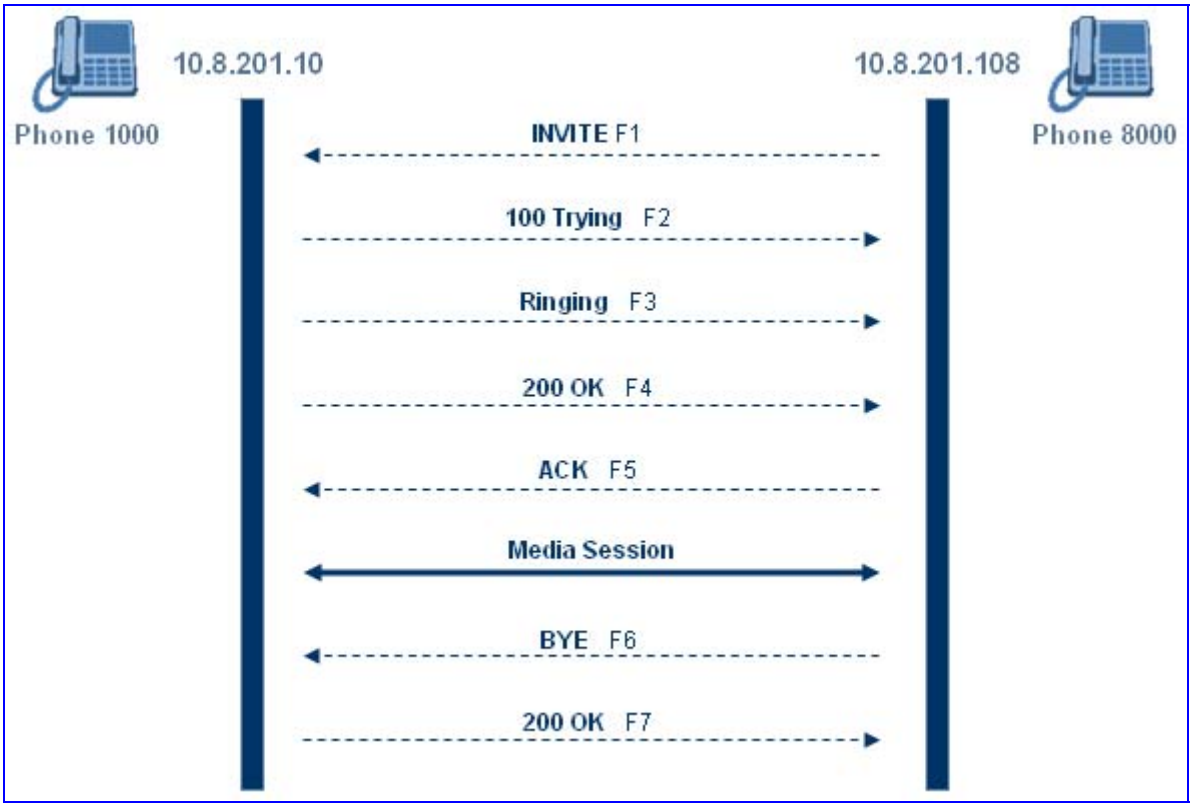
If registration per B-channel is selected, on device startup the device sends REGISTER requests according to the maximum number of allowed SIP dialogs (configured by the parameter NumberOfActiveDialogs). After each received response, the subsequent REGISTER request is sent.

7.13 Configuration Examples

7.13.1 SIP Call Flow

The SIP call flow (shown in the following figure), describes SIP messages exchanged between two devices during a simple call. In this call flow example, device (10.8.201.158) with phone number '6000' dials device (10.8.201.161) with phone number '2000'.

Figure 7-4: SIP Call Flow



- **F1 (10.8.201.108 >> 10.8.201.10 INVITE):**

```
INVITE sip:1000@10.8.201.10;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacsiJkDGd
From: <sip:8000@10.8.201.108>;tag=1c5354
To: <sip:1000@10.8.201.10>
Call-ID: 534366556655skKw-8000--1000@10.8.201.108
CSeq: 18153 INVITE
Contact: <sip:8000@10.8.201.108;user=phone>
User-Agent: Audiocodes-Sip-Gateway/Mediant 1000/v.5.40.010.006
Supported: 100rel,em
Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,
NOTIFY,PRACK,REFER,INFO
Content-Type: application/sdp
Content-Length: 208
v=0
o=AudiocodesGW 18132 74003 IN IP4 10.8.201.108
s=Phone-Call
c=IN IP4 10.8.201.108
t=0 0
m=audio 4000 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
```

- **F2 (10.8.201.10 >> 10.8.201.108 TRYING):**

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacsiJkDGd
From: <sip:8000@10.8.201.108>;tag=1c5354
To: <sip:1000@10.8.201.10>
Call-ID: 534366556655skKw-8000--1000@10.8.201.108
Server: Audiocodes-Sip-Gateway/Mediant 1000/v.5.40.010.006
CSeq: 18153 INVITE
Content-Length: 0
```

- **F3 (10.8.201.10 >> 10.8.201.108 180 RINGING):**

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacsiJkDGd
From: <sip:8000@10.8.201.108>;tag=1c5354
To: <sip:1000@10.8.201.10>;tag=1c7345
Call-ID: 534366556655skKw-8000--1000@10.8.201.108
Server: Audiocodes-Sip-Gateway/Mediant 1000/v.5.40.010.006
CSeq: 18153 INVITE
Supported: 100rel,em
Content-Length: 0
```

Note: Phone '1000' answers the call and then sends a 200 OK message to device 10.8.201.108.

- **F4 (10.8.201.10 >> 10.8.201.108 200 OK):**

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacsiJkDGd
From: <sip:8000@10.8.201.108>;tag=1c5354
To: <sip:1000@10.8.201.10>;tag=1c7345
Call-ID: 534366556655skKw-8000--1000@10.8.201.108
CSeq: 18153 INVITE
Contact: <sip:1000@10.8.201.10;user=phone>
Server: Audiocodes-Sip-Gateway/Mediant 1000/v.5.40.010.006
Supported: 100rel,em
Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,
NOTIFY,PRACK,REFER,INFO
Content-Type: application/sdp
Content-Length: 206
v=0
o=AudiocodesGW 30221 87035 IN IP4 10.8.201.10
s=Phone-Call
c=IN IP4 10.8.201.10
t=0 0
m=audio 7210 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=ptime:20
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
```

- **F5 (10.8.201.108 >> 10.8.201.10 ACK):**

```
ACK sip:1000@10.8.201.10;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacZYpJWxZ
From: <sip:8000@10.8.201.108>;tag=1c5354
To: <sip:1000@10.8.201.10>;tag=1c7345
Call-ID: 534366556655skKw-8000--1000@10.8.201.108
User-Agent: Audiocodes-Sip-Gateway/Mediant 1000/v.5.40.010.006
CSeq: 18153 ACK
Supported: 100rel,em
Content-Length: 0
```

Note: Phone '8000' goes on-hook and device 10.8.201.108 sends a BYE to device 10.8.201.10. Voice path is established.

- **F6 (10.8.201.108 >> 10.8.201.10 BYE):**

```
BYE sip:1000@10.8.201.10;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacRKCVBud
From: <sip:8000@10.8.201.108>;tag=1c5354
To: <sip:1000@10.8.201.10>;tag=1c7345
Call-ID: 534366556655skKw-8000--1000@10.8.201.108
User-Agent: Audiocodes-Sip-Gateway/Mediant 1000/v.5.40.010.006
CSeq: 18154 BYE
Supported: 100rel,em
Content-Length: 0
```

- **F7 (10.8.201.10 >> 10.8.201.108 200 OK):**

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacRKCVBud
From: <sip:8000@10.8.201.108>;tag=1c5354
To: <sip:1000@10.8.201.10>;tag=1c7345
Call-ID: 534366556655skKw-8000--1000@10.8.201.108
Server: Audiocodes-Sip-Gateway/Mediant 1000/v.5.40.010.006
CSeq: 18154 BYE
Supported: 100rel,em
Content-Length: 0
```

7.13.2 SIP Authentication Example

The device supports basic and digest (MD5) authentication types, according to SIP RFC 3261 standard. A proxy server might require authentication before forwarding an INVITE message. A Registrar/Proxy server may also require authentication for client registration. A proxy replies to an unauthenticated INVITE with a 407 Proxy Authorization Required response, containing a Proxy-Authenticate header with the form of the challenge. After sending an ACK for the 407, the user agent can then resend the INVITE with a Proxy-Authorization header containing the credentials.

User agent, redirect or registrar servers typically use 401 Unauthorized response to challenge authentication containing a WWW-Authenticate header, and expect the re-INVITE to contain an Authorization header.

The following example describes the Digest Authentication procedure, including computation of user agent credentials:

1. The REGISTER request is sent to Registrar/Proxy server for registration, as follows:

```
REGISTER sip:10.2.2.222 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c17940
To: <sip: 122@10.1.1.200>
Call-ID: 634293194@10.1.1.200
User-Agent: Audiocodes-Sip-Gateway/Mediant 1000/v.5.40.010.006
CSeq: 1 REGISTER
Contact: sip:122@10.1.1.200:
Expires:3600
```

2. Upon receipt of this request, the Registrar/Proxy returns 401 Unauthorized response.

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 10.2.1.200
From: <sip:122@10.2.2.222 >;tag=1c17940
To: <sip:122@10.2.2.222 >
Call-ID: 634293194@10.1.1.200
Cseq: 1 REGISTER
Date: Mon, 30 Jul 2001 15:33:54 GMT
Server: Columbia-SIP-Server/1.17
Content-Length: 0
WWW-Authenticate: Digest realm="audiocodes.com",
nonce="11432d6bce58ddf02e3b5e1c77c010d2",
stale=FALSE,
algorithm=MD5
```

3. According to the sub-header present in the WWW-Authenticate header, the correct REGISTER request is formed.

4. Since the algorithm is MD5, then:
   - The username is equal to the endpoint phone number 122.
   - The realm return by the proxy is audiocodes.com.
   - The password from the *ini* file is AudioCodes.
   - The equation to be evaluated is (according to RFC this part is called A1) **'122:audiocodes.com:AudioCodes'.**
   - The MD5 algorithm is run on this equation and stored for future usage.
   - The result is 'a8f17d4b41ab8dab6c95d3c14e34a9e1'.
5. Next, the par called A2 needs to be evaluated:
   - The method type is 'REGISTER'.
   - Using SIP protocol 'sip'.
   - Proxy IP from *ini* file is '10.2.2.222'.
   - The equation to be evaluated is **'REGISTER:sip:10.2.2.222'.**
   - The MD5 algorithm is run on this equation and stored for future usage.
   - The result is 'a9a031cfddcb10d91c8e7b4926086f7e'.
6. Final stage:
   - The A1 result: The nonce from the proxy response is '11432d6bce58ddf02e3b5e1c77c010d2'.
   - The A2 result: The equation to be evaluated is **'A1:11432d6bce58ddf02e3b5e1c77c010d2:A2'.**
   - The MD5 algorithm is run on this equation. The outcome of the calculation is the response needed by the device to register with the Proxy.
   - The response is 'b9c45d0234a5abf5ddf5c704029b38cf'.

   At this time, a new REGISTER request is issued with the following response:

```
REGISTER sip:10.2.2.222 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c23940
To: <sip: 122@10.1.1.200>
Call-ID: 654982194@10.1.1.200
Server: Audiocodes-Sip-Gateway/Mediant 1000/v.5.40.010.006
CSeq: 1 REGISTER
Contact: sip:122@10.1.1.200:
Expires:3600
Authorization: Digest, username: 122,
realm="audiocodes.com",
nonce="11432d6bce58ddf02e3b5e1c77c010d2",
uri="10.2.2.222",
response="b9c45d0234a5abf5ddf5c704029b38cf"
```

7. Upon receiving this request and if accepted by the Proxy, the proxy returns a 200 OK response closing the REGISTER transaction:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c23940
To: <sip: 122@10.1.1.200>
Call-ID: 654982194@10.1.1.200
Cseq: 1 REGISTER
Date: Thu, 26 Jul 2001 09:34:42 GMT
Server: Columbia-SIP-Server/1.17
Content-Length: 0
Contact: <sip:122@10.1.1.200>; expires="Thu, 26 Jul 2001 10:34:42
GMT"; action=proxy; q=1.00
Contact: <122@10.1.1.200:>; expires="Tue, 19 Jan 2038 03:14:07
GMT"; action=proxy; q=0.00
Expires: Thu, 26 Jul 2001 10:34:42 GMT
```

7.13.3 Establishing a Call between Two Devices

This section provides an example on configuring two AudioCodes' devices with FXS interfaces for establishing call communication. After configuration, you can make calls between telephones connected to the same device or between the two devices.

Note: This section is applicable only to Mediant 1000.

In the example, the IP address of the first device is 10.2.37.10 and its endpoint numbers are 101 to 104. The IP address of the second device is 10.2.37.20 and its endpoint numbers are 201 to 204. In this example, a SIP Proxy is not used. Internal call routing is performed using the device's 'Tel to IP Routing' table.

➢ To configure the two devices for call communication, take these 4 steps:

1. For the *first* device (10.2.37.10), in the 'Trunk Group Table' page (refer to "Configuring the Trunk Group Table" on page 226), assign the phone numbers 101 to 104to the device's endpoints.

Figure 7-5: Assigning Phone Numbers to Device 10.2.37.10)

Group Index	Module	From Trunk	To Trunk	Channels	Phone Number	Trunk Group ID
1	Module 3 FXS	1	1	1-4	101	0

2. For the *second* device (10.2.37.20), in the 'Trunk Group Table' page, assign the phone numbers 201 to 204to the device's endpoints.

Figure 7-6: Assigning Phone Numbers to Device 10.2.37.20

Group Index	Module	From Trunk	To Trunk	Channels	Phone Number	Trunk Group ID
1	Module 3 FXS	1	1	1-4	201	0

3. Configure the following settings for *both* devices:

   In the 'Tel to IP Routing' page (refer to "Tel to IP Routing Table" on page 200), add the following routing rules:

   a. In the first row, enter 10 for the destination phone prefix and enter 10.2.37.10 for the destination IP address (i.e., IP address of the first device).

   b. In the second row, enter 20 for the destination phone prefix and 10.2.37.20 for the destination IP address (i.e., IP address of the second device).

   These settings enable the routing (from both devices) of outgoing Tel-to-IP calls that start with 10 to the first device and calls that start with 20 to the second device.

Figure 7-7: Routing Calls Between Devices

	Src. Trunk Group ID	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address	Dest. IP Group ID
1		10	*		10.2.37.10	
2		20	*		10.2.37.20	

4. Make a call. Pick up the phone connected to port #1 of the first device and dial 102 (to the phone connected to port #2 of the same device). Listen for progress tones at the calling phone and for the ringing tone at the called phone. Answer the called phone, speak into the calling phone, and check the voice quality. Dial 201 from the phone connected to port #1 of the first device; the phone connected to port #1 of the second device rings. Answer the call and check the voice quality.

7.13.4 Remote PBX Extension Between FXO and FXS Devices

Remote PBX extension offers a company the capability of extending the "power" of its local PBX by allowing remote phones (remote offices) to connect to the company's PBX over the IP network (instead of via PSTN). This is as if the remote office is located in the head office (where the PBX is installed). PBX extensions are connected through FXO ports to the IP network, instead of being connected to individual telephone stations. At the remote office, FXS units connect analog phones to the same IP network. To produce full transparency, each FXO port is mapped to an FXS port (i.e., one-to-one mapping). This allows individual extensions to be extended to remote locations. To call a remote office worker, a PBX user or a PSTN caller simply dials the PBX extension that is mapped to the remote FXS port.

This section provides an example on how to implement a remote telephone extension through the IP network, using FXO andFXS interfaces (i.e., modules). In this configuration, the FXO device routes calls received from the PBX to the 'Remote PBX Extension' connected to the FXS device. The routing is transparent as if the telephone connected to the FXS device is directly connected to the PBX.

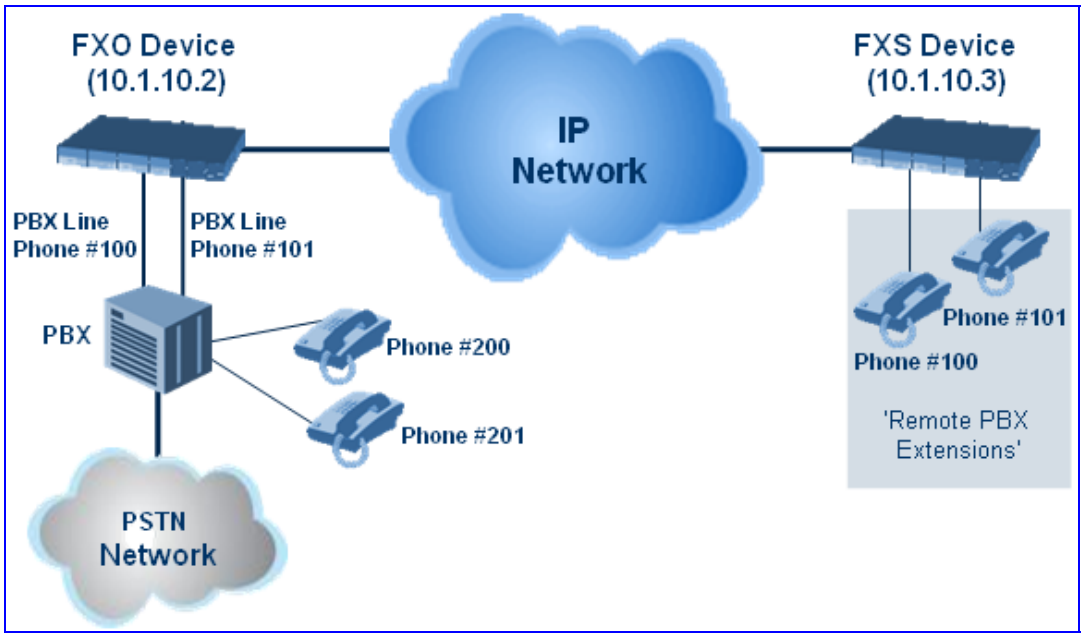
Note: This section is applicable only to Mediant 1000.

The following is required:

- FXO interfaces with ports connected directly to the PBX lines (shown in the figure below)
- FXS interfaces for the 'remote PBX extension'

- Analog phones (POTS)
- PBX (one or more PBX loop start lines)
- LAN network

Figure 7-8: FXO-FXS Remote PBX Extension (Example)



7.13.4.1 Dialing from Remote Extension (Phone at FXS)

The procedure below describes how to dial from the 'remote PBX extension' (i.e., phone connected to the FXS interface).

➢ To make a call from the FXS interface, take these 3 steps:

1. Off-hook the phone and wait for the dial tone from the PBX. This is as if the phone is connected directly to the PBX. The FXS and FXO interfaces establish a voice path connection from the phone to the PBX immediately after the phone is off-hooked.
2. Dial the destination number (e.g., phone number 201). The DTMF digits are sent over IP directly to the PBX. All the audible tones are generated from the PBX (such as ringback, busy, or fast busy tones). One-to-one mapping occurs between the FXS ports and PBX lines.
3. The call disconnects when the phone connected to the FXS goes on-hook.

7.13.4.2 Dialing from PBX Line or PSTN

The procedure below describes how to dial from a PBX line (i.e., from a telephone directly connected to the PBX) or from the PSTN to the 'remote PBX extension' (i.e., telephone connected to the FXS interface).

➢ **To dial from a telephone directly connected to the PBX or from the PSTN, take this step:**

- Dial the PBX subscriber number (e.g., phone number 101) in the same way as if the user's phone was connected directly to the PBX. As soon as the PBX rings the FXO device, the ring signal is 'sent' to the phone connected to the FXS device. Once the phone connected to the FXS device is off-hooked, the FXO device seizes the PBX line and the voice path is established between the phone and PBX.

  There is one-to-one mapping between PBX lines and FXS device ports. Each PBX line is routed to the same phone (connected to the FXS device). The call disconnects when the phone connected to the FXS device is on-hooked.

7.13.4.3 Message Waiting Indication for Remote Extensions

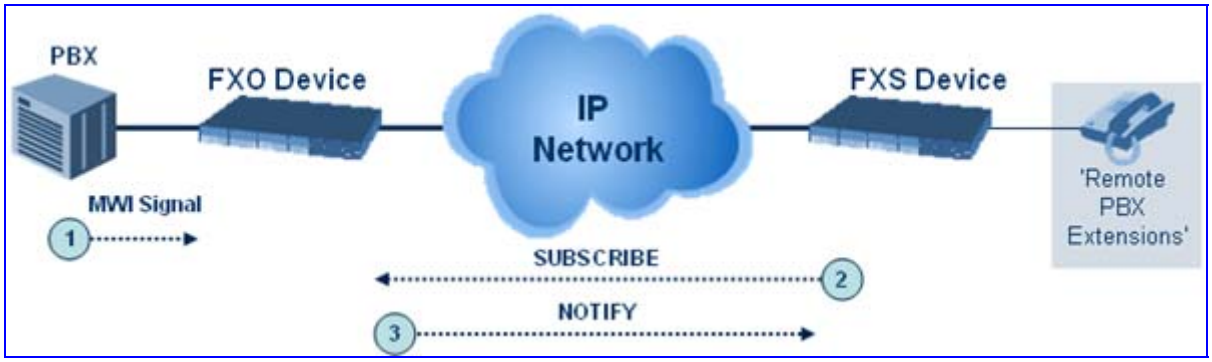
The device supports the relaying of Message Waiting Indications (MWI) for remote extensions (and voice mail applications). Instead of subscribing to an MWI server to receive notifications of pending messages, the FXO device receives subscriptions from the remote FXS device and notifies the appropriate extension when messages (and the number of messages) are pending.

The FXO device detects an MWI message from the Tel (PBX) side using any one of the following methods:

- 100 VDC (sent by the PBX to activate the phone's lamp)
- Stutter dial tone from the PBX
- MWI display signal (according to the parameter CallerIDType)

Upon detection of an MWI message, the FXO device sends a SIP NOTIFY message to the IP side. When receiving this NOTIFY message, the remote FXS device generates an MWI signal toward its Tel side.

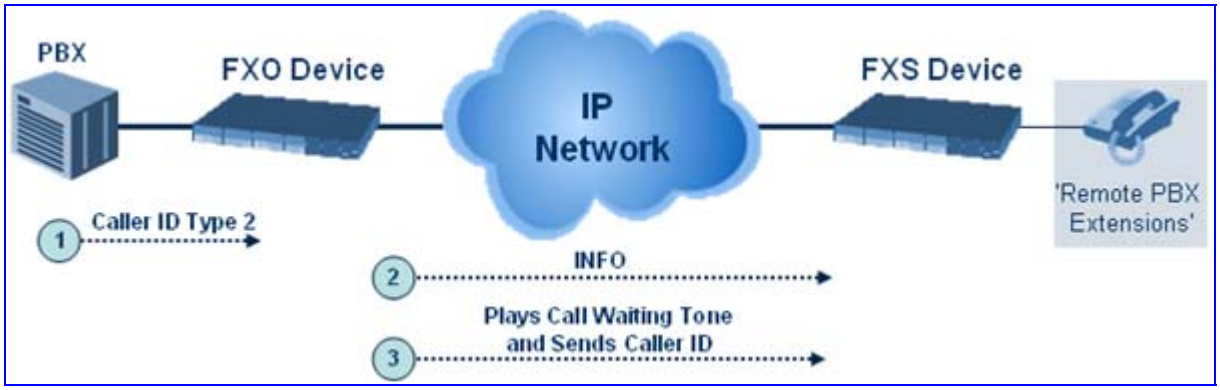
Figure 7-9: MWI for Remote Extensions



7.13.4.4 Call Waiting for Remote Extensions

When the FXO device detects a Call Waiting indication (FSK data of the Caller Id - CallerIDType2) from the PBX, it sends a proprietary INFO message, which includes the caller identification to the FXS device. Once the FXS device receives this INFO message, it plays a call waiting tone and sends the caller ID to the relevant port for display. The remote extension connected to the FXS device can toggle between calls using the Hook Flash button.

Figure 7-10: Call Waiting for Remote Extensions



7.13.4.5 FXS Gateway Configuration

The procedure below describes how to configure the FXS interface (at the 'remote PBX extension').

➢ To configure the FXS interface, take these 3 steps:

1. In the 'Trunk Group Table' page (refer to "Configuring the Trunk Group Table" on page 226, assign the phone numbers 100 to 104 to the device's endpoints.

Figure 7-11: Assigning Phone Numbers to FXS Endpoints

Group Index	Module	From Trunk	To Trunk	Channels	Phone Number	Trunk Group ID
1	Module 3 FXS	1	1	1-4	100	0

2. In the 'Automatic Dialing' page (refer to "Automatic Dialing" on page 220), enter the phone numbers of the FXO device in the 'Destination Phone Number' fields. When a phone connected to Port #1 off-hooks, the FXS device automatically dials the number '200'.

Figure 7-12: Automatic Dialing for FXS Ports

Gateway Port	Destination Phone Number	Auto Dial Status
Module 3 Port 1 FXS	200	Enable
Module 3 Port 2 FXS	201	Enable
Module 3 Port 3 FXS	202	Enable
Module 3 Port 4 FXS	203	Enable

3. In the 'Tel to IP Routing' page (refer to "Tel to IP Routing Table" on page 200), enter 20 for the destination phone prefix, and 10.1.10.2 for the IP address of the FXO device.

	Src. Trunk Group ID	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address
1		20	*		10.1.10.2

Note: For the transfer to function in remote PBX extensions, Hold must be disabled at the FXS device (i.e., Enable Hold = 0) and hook-flash must be transferred from the FXS to the FXO (HookFlashOption = 4).

7.13.4.6 FXO Gateway Configuration

The procedure below describes how to configure the FXO interface (to which the PBX is directly connected).

➢ To configure the FXO interface, take these 4 steps:

1. In the 'Endpoint Phone Numbers' page, assign the phone numbers 200 to 207 to the device's FXO endpoints.

Figure 7-13: Assigning Phone Numbers to FXO Ports

Group Index	Module	From Trunk	To Trunk	Channels	Phone Number
1	Module 4 FXO	1	1	1-4	200

2. In the 'Automatic Dialing' page, enter the phone numbers of the FXS device in the 'Destination Phone Number' fields. When a ringing signal is detected at Port #1, the FXO device automatically dials the number '100'.

Figure 7-14: FXO Automatic Dialing Configuration

Gateway Port	Destination Phone Number	Auto Dial Status
Module 3 Port 1 FXO	100	Enable
Module 3 Port 2 FXO	101	Enable
Module 3 Port 3 FXO	102	Enable
Module 3 Port 4 FXO	103	Enable

3. In the 'Tel to IP Routing' page, enter 10 in the 'Destination Phone Prefix' field, and the IP address of the FXS device (10.1.10.3) in the field 'IP Address'.

Figure 7-15: FXO Tel-to-IP Routing Configuration

	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address
1	10	*		10.1.10.3

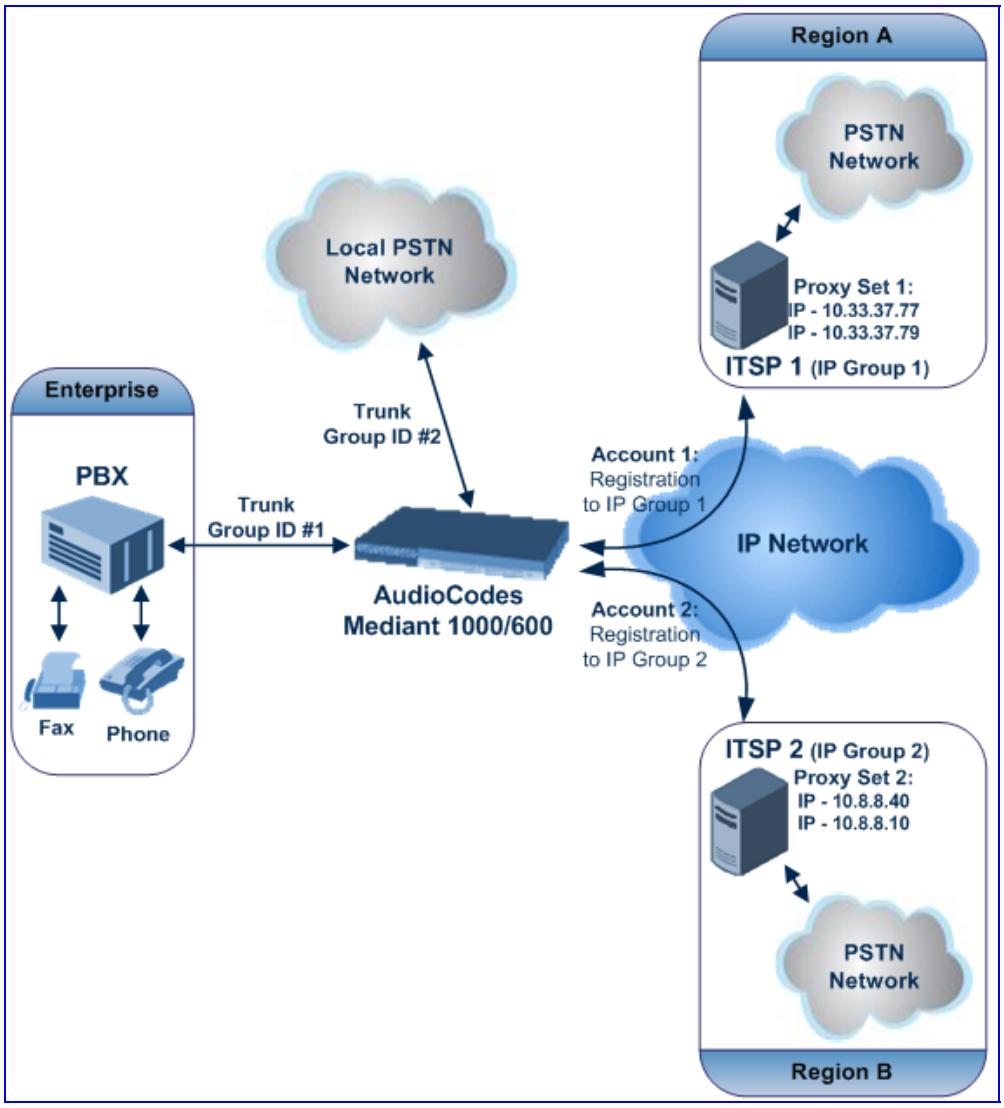
4. In the 'FXO Settings' page (refer to "Configuring the FXO Parameters" on page 248), set the parameter 'Dialing Mode' to 'Two Stages' (IsTwoStageDial = 1).

7.13.5 SIP Trunking between Enterprise and ITSPs

By implementing the device's enhanced and flexible routing configuration capabilities using Proxy Sets, IP Groups, and Accounts, you can "design" complex routing schemes. This section provides an example of an elaborate routing scheme for implementing SIP trunking between an Enterprise's PBX and two Internet Telephony Service Providers (ITSP), using AudioCodes device.

Scenario: In this example, the Enterprise wishes to connect its TDM PBX to two different ITSPs, by implementing a device in its network environment. It's main objective is for the device to route Tel-to-IP calls to these ITSPs according to a dial plan. The device is to register (on behalf of the PBX) to each ITSP, which implements two servers for redundancy and load balancing. The Register messages are to use different URI's in the From, To, and Contact headers per ITSP. In addition, all calls dialed from the Enterprise PBX with prefix '02' is sent to the local PSTN. The figure below illustrates the example setup:

Figure 7-16: Example Setup for Routing Between ITSP and Enterprise PBX

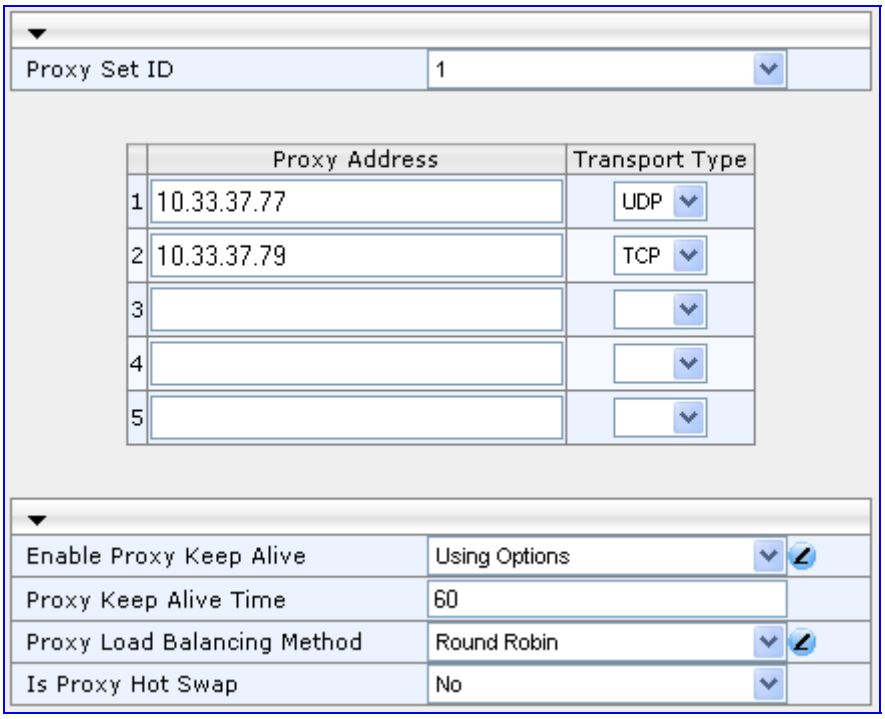


➢ To configure call routing between Enterprise and two ITSPs using the device, take these 8 steps:

1. Enable the device to register to a Proxy / Registrar server, using the parameter IsRegisterNeeded in the 'Proxy & Registration' page (refer to "Proxy & Registration Parameters" on page 146).
2. In the 'Proxy Sets Table' page (refer to "Proxy Sets Table" on page 153), configure two Proxy Sets and for each, enable Proxy Keep-Alive (using SIP OPTIONS) and 'round robin' load-balancing method:
   - Proxy Set **#1** includes two IP addresses of the first ITSP (**ITSP 1**) - 10.33.37.77 and 10.33.37.79 - and using UDP.
   - Proxy Set **#2** includes two IP addresses of the second ITSP (**ITSP 2**) - 10.8.8.40 and 10.8.8.10 - and using TCP.

   The figure below displays the configuration of Proxy Set ID #1. Perform similar configuration for Proxy Set ID #2, but using different IP addresses.

Figure 7-17: Configuring Proxy Set ID #1 in the Proxy Sets Table Page



3. In the 'IP Group Table' page (refer to "Configuring the IP Groups" on page 231), configure the two IP Groups IDs #1 and #2. Assign Proxy Sets #1 and #2 to IP Groups #1 and #2 respectively.

Figure 7-18: Configuring IP Groups #1 and #2 in the IP Group Table Page

IP Group ID	Description	Proxy Set ID	SIP Group Name	Send Invite To Proxy	Always Use Route Table
1	ITSP_1	1		Disable	Disable
2	ITSP_2	2		Disable	Disable

4. In the 'Trunk Group Table' page (refer to "Configuring the Trunk Group Table" on page 226), enable the Trunks connected between the Enterprise's PBX and the device (Trunk Group ID #1), and between the local PSTN and the device (Trunk Group ID #2).

Figure 7-19: Assign the Trunk to Trunk Group ID #1 in the Trunk Group Table Page

Group Index	From Trunk	To Trunk	Channels	Phone Number	Trunk Group ID	IP Profile ID
1	1	1	1-31	1100	1	0
2	2	2	1-31	2200	2	0

5. In the 'Trunk Group Settings' page (refer to "Configuring the Trunk Group Settings" on page 228), configure 'Per Account' registration for Trunk Group ID #1 (without serving IP Group).

Figure 7-20: Configuring Trunk Group #1 for Registration in Trunk Group Settings Page

	Trunk Group ID	Channel Select Mode	Registration Mode	Serving IP Group ID	Gateway Name	Contact User
1	1	Cyclic Ascending	Per Account			username

6. In the 'Account Table' page (refer to "Configuring the Account Table" on page 232), configure the two Accounts for PBX trunk registration to ITSPs using the same Trunk Group (i.e., ID #1), but different serving IP Groups #1 and #2. For each account, define user name, password, and hostname, and ContactUser. The Register messages use different URI's (Hostname and ContactUser) in the From, To, and Contact headers per ITSP. Enable registration for both accounts.

Figure 7-21: Configuring Accounts for PBX Registration to ITSPs in Account Table Page

Index	ServedTrunkGroup	ServingIPGroup	Username	Password	HostName	Register	ContactUser
1	1	1	user1	1234	ITSP1	1	ITSP1user
2	1	2	user2	5555	ITSP2	1	ITSP2user

7. In the 'IP to Trunk Group Routing' page (refer to "IP to Trunk Group Routing" on page 204), configure IP-to-Tel routing for calls from ITSPs to Trunk Group ID #1 (see 1 below) and from the device to the local PSTN (see 2 below).

Figure 7-22: Configuring ITSP-to-Trunk Group #1 Routing in IP to Trunk Group Table Page

	Dest. Host Prefix	Source Host Prefix	Dest. Phone Prefix	Source Phone Prefix	Source IP Address	->	Trunk Group ID	IP Profile ID	Source IPGroup ID
1			*	*			1		
2			02	*			2		

8. In the 'Tel to IP Routing' page (refer to "Tel to IP Routing Table" on page 200), configure Tel-to-IP routing rules for calls to ITSPs (see 1 below) and to local PSTN (see 2 below) .

Figure 7-23: Configuring Tel-to-IP Routing to ITSPs in Tel to IP Routing Table Page

	Src. Trunk Group ID	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address	Dest. IP Group ID	IP Profile ID
1	1	0[3,4,5]	*			1	
2	1	0[6,7,8]	*			2	
3	1	02	*		10.13.4.13		

7.14 Working with Supplementary Services

The device supports the following supplementary services:

- Call Hold and Retrieve (refer to "Call Hold and Retrieve" on page 437).
- Consultation / Alternate (refer to "Consultation / Alternate" on page 438).
- Call Transfer (refer to "Call Transfer" on page 438).
- Call Forward: analog interface - 3xx Redirect Responses; digital interface - when a callRerouting IE is received in a FACILITY message in response to an outgoing SETUP message, the device sends a 3xx response to the IP side, including the callRerouting destination number - only applicable to QSIG protocol (refer to "Call Forward" on page 439).
- Call Waiting - 182 Queued Response (refer to "Call Waiting" on page 440).
- Message Waiting Indication - MWI (refer to "Message Waiting Indication" on page 440).
- Caller ID (refer to "Caller ID" on page 441).

To activate these supplementary services (Hold, Transfer, Forward, Waiting and MWI) on the device, enable each service's corresponding parameter either from the Web interface or via the ini file.

Notes:

- All call participants must support the specific supplementary service that is used.
- When working with certain application servers (such as BroadSoft's BroadWorks) in client server mode (the application server controls all supplementary services and keypad features by itself), the device's supplementary services must be disabled.

7.14.1 Call Hold and Retrieve

Initiating Hold / Retrieve:

- Active calls can be put on-hold by pressing the phone's hook-flash button.
- The party that initiates the hold is called the *holding* party; the other party is called the *held* party.
- After a successful Hold, the holding party hears a Dial Tone (HELD_TONE, defined in the device's Call Progress Tones file).
- Call retrieve can be performed only by the holding party while the call is held and active.
- The holding party performs the retrieve by pressing the hook-flash.
- After a successful retrieve, the voice is connected again.
- Hold is performed by sending Re-INVITE message with IP address 0.0.0.0 or a=sendonly in the SDP according to the parameter HoldFormat.

- The hold and retrieve functionalities are implemented by Re-INVITE messages. The IP address 0.0.0.0 as the connection IP address or the string 'a=inactive' in the received Re-INVITE SDP cause the device to enter Hold state and to play held tone (configured in the device) to the PBX/PSTN. If the string 'a=sendonly' is received in the SDP message, the device stops sending RTP packets, but continues to listen to the incoming RTP packets. Usually, the remote party plays, in this scenario, Music on Hold (MOH) and the device forwards the MOH to the held party.

Receiving Hold / Retrieve:

- When an active call receives Re-INVITE message with either the IP address 0.0.0.0 or the 'inactive' string in SDP, the device stops sending RTP and plays a local Held Tone.
- When an active call receives Re-INVITE message with 'sendonly' string in SDP, the device stops sending RTP and listens to the remote party. In this mode, it is expected that on-hold music (or any other hold tone) is to be played (over IP) by the remote party.

7.14.2 Consultation / Alternate

- The consultation feature is relevant only for the holding party (applicable only to the FXS module).
- After holding a call (by pressing hook-flash), the holding party hears a dial tone and can now initiate a new call, which is called a consultation call.
- While hearing a dial tone, or when dialing to the new destination (before dialing is complete), the user can retrieve the held call by pressing hook-flash.
- The held call can't be retrieved while Ringback tone is heard.
- After the consultation call is connected, the user can switch between the held and active call by pressing hook-flash.

Note: This section is applicable only to Mediant 1000.

7.14.3 Call Transfer

There are two types of call transfers:

- **Consultation Transfer** (REFER and REPLACES)**:**

  The common way to perform a consultation transfer is as follows:

  In the transfer scenario there are three parties: Party A = transferring, Party B = transferred, Party C = transferred to.

  - A Calls B.
  - B answers.
  - A presses the hook-flash and puts B on-hold (party B hears a hold tone).
  - A dials C.
  - After A completes dialing C, A can perform the transfer by on-hooking the A phone.
  - After the transfer is complete, B and C parties are engaged in a call.

The transfer can be initiated at any of the following stages of the call between A and C:

- Just after completing dialing C phone number - transfer from setup.
- While hearing Ringback – transfer from alert.
- While speaking to C - transfer from active.

- **Blind Transfer** (REFER)**:**

  Blind transfer is performed after we have a call between A and B, and party A decides to immediately transfer the call to C without speaking with C. The result of the transfer is a call between B and C (just like consultation transfer only skipping the consultation stage).

Note the following SIP issues:

- Transfer is initiated by sending REFER with REPLACES.
- The device can receive and act upon receiving REFER with or without REPLACES.
- The device can receive and act upon receiving INVITE with REPLACES, in which case the old call is replaced by the new one.
- The INVITE with REPLACES can be used to implement Directed Call Pickup.

7.14.4 Call Forward

The following forms of call forward are supported:

- **Immediate:** incoming call is forwarded immediately and unconditionally.
- **Busy:** incoming call is forwarded if the endpoint is busy.
- **No Reply:** incoming call is forwarded if it isn't answered for a specified time.
- **On Busy or No Reply:** incoming call is forwarded if the port is busy or when calls are not answered after a specified time.
- **Do Not Disturb:** immediately reject incoming calls. Upon receiving a call to Do Not Disturb call, the 603 Decline SIP response code is sent.

Three forms of forwarding parties are available:

- **Served party:** party configured to forward the call (FXS device).
- **Originating party:** party that initiated the first call (FXS or FXO device).
- **Diverted party:** new destination of the forwarded call (FXS or FXO device).

The served party (FXS interface) can be configured through the Web interface (refer to "Call Forward" on page 222) or *ini* file to activate one of the call forward modes. These modes are configurable per device's endpoints.

Notes:

- This section is applicable only to Mediant 1000.
- When call forward is initiated, the device sends a SIP 302 response with a contact that contains the phone number from the forward table and its corresponding IP address from the routing table (or when a proxy is used, the proxy's IP address).
- For receiving call forward, the device handles SIP 3xx responses for redirecting calls with a new contact.

7.14.5 Call Waiting

The Call Waiting feature enables FXS devices to accept an additional (second) call on busy endpoints. If an incoming IP call is designated to a busy port, the called party hears call waiting tone (several configurable short beeps) and (for Bellcore and ETSI Caller IDs) can view the Caller ID string of the incoming call. The calling party hears a Call Waiting Ringback Tone. The called party can accept the new call using hook-flash, and can toggle between the two calls.

Note: This section is applicable only to Mediant 1000.

To enable Call Waiting:

- Set EnableCallWaiting = 1 (or using the Web interface, refer to "Supplementary Services" on page 176)
- Set EnableHold = 1.
- Define the Call Waiting indication and Call Waiting Ringback tones in the Call Progress Tones file. You can define up to four Call Waiting indication tones (refer to the parameter FirstCallWaitingToneID in "SIP Configuration Parameters" on page 323).
- To configure the Call Waiting indication tone cadence, modify the following parameters: NumberOfWaitingIndications, WaitingBeepDuration and TimeBetweenWaitingIndications (or using the Web interface, refer to "Supplementary Services" on page 176).
- To configure a delay interval before a Call Waiting Indication is played to the currently busy port use the parameter TimeBeforeWaitingIndication (or using the Web interface, refer to "Supplementary Services" on page 176). This enables the caller to hang up before disturbing the called party with Call Waiting Indications. Applicable only to FXS modules.

Both the calling and called sides are supported by FXS modules; the FXO modules support only the calling side.

To indicate Call Waiting, the device sends a 182 Call Queued response. The device identifies a Waiting Call when a 182 Call Queued response is received.

7.14.6 Message Waiting Indication

Support for Message Waiting Indication (MWI) according to IETF <draft-ietf-sipping-mwi-04.txt>, including SUBSCRIBE (to MWI server). The FXS device can accept an MWI NOTIFY message that indicates waiting messages or that the MWI is cleared. Users are informed of these messages by a stutter dial tone. The stutter and confirmation tones are defined in the CPT file (refer to Configuring the Call Progress Tones File in the *Product Reference Manual*). If the MWI display is configured, the number of waiting messages is also displayed. If the MWI lamp is configured, the phone's lamp (on a phone that is equipped with an MWI lamp) is lit. The device can subscribe to the MWI server per port (usually used on FXS) or per device (used on FXO).

Note: This section is applicable only to Mediant 1000.

To configure MWI, set the following parameters:

- EnableMWI (or using the Web interface, refer to "Supplementary Services" on page 176)
- MWIServerIP (or using the Web interface, refer to "Supplementary Services" on page 176)
- MWIAnalogLamp (or using the Web interface, refer to "Supplementary Services" on page 176)
- MWIDisplay (or using the Web interface, refer to "Supplementary Services" on page 176)
- StutterToneDuration (or using the Web interface, refer to "Supplementary Services" on page 176)
- EnableMWISubscription (or using the Web interface, refer to "Supplementary Services" on page 176)
- MWIExpirationTime (or using the Web interface, refer to "Supplementary Services" on page 176)
- SubscribeRetryTime (or using the Web interface, refer to "Supplementary Services" on page 176)
- SubscriptionMode (or using the Web interface, refer to "Proxy & Registration Parameters" on page 146)
- CallerIDType -- determines the standard for detection of MWI signals (or using the Web interface, refer to "Supplementary Services" on page 176)
- ETSIVMWITypeOneStandard (for a description, refer to "Analog Telephony Parameters" on page 355)
- BellcoreVMWITypeOneStandard (for a description, refer to "Analog Telephony Parameters" on page 355)

7.14.7 Caller ID

This section discusses the device's Caller ID support for analog modules.

Note: This section is applicable only to Mediant 1000.

7.14.7.1 Caller ID Detection / Generation on the Tel Side

By default, generation and detection of Caller ID to the Tel side is disabled. To enable Caller ID, set the parameter EnableCallerID to 1. When the Caller ID service is enabled:

- For FXS: the Caller ID signal is sent to the device's port
- For FXO: the Caller ID signal is detected

The configuration for Caller ID is described below:

- Use the parameter CallerIDType to define the Caller ID standard. Note that the Caller ID standard that is used on the PBX or phone must match the standard defined in the device.
- Select the Bellcore caller ID sub standard using the parameter BellcoreCallerIDTypeOneSubStandard
- Select the ETSI FSK caller ID sub standard using the parameter ETSICallerIDTypeOneSubStandard
- Enable or disable (per port) the caller ID generation (for FXS) and detection (for FXO) using the 'Generate / Detect Caller ID to Tel' table (EnableCallerID). If a port isn't configured, its caller ID generation / detection are determined according to the global parameter EnableCallerID.
- EnableCallerIDTypeTwo: disables / enables the generation of Caller ID type 2 when the phone is off-hooked (used for call waiting).
- RingsBeforeCallerID: sets the number of rings before the device starts detection of caller ID (FXO only). By default, the device detects the caller ID signal between the first and second rings.
- AnalogCallerIDTimimgMode: determines the time period when a caller ID signal is generated (FXS only). By default, the caller ID is generated between the first two rings.
- PolarityReversalType: some Caller ID signals use reversal polarity and/or wink signals. In these scenarios, it is recommended to set PolarityReversalType to 1 (Hard) (FXS only).
- The Caller ID interworking can be changed using the parameters UseSourceNumberAsDisplayName and UseDisplayNameAsSourceNumber.

7.14.7.2 Debugging a Caller ID Detection on FXO

➢ To debug a Caller ID detection on an FXO interface, take these 7 steps:

1. Verify that the parameter EnableCallerID is set to 1.
2. Verify that the caller ID standard (and substandard) of the device matches the standard of the PBX (CallerIDType, BellcoreCallerIDTypeOneSubStandard, and ETSICallerIDTypeOneSubStandard).
3. Define the number of rings before the device starts detection of caller ID (RingsBeforeCallerID).
4. Verify that the coefficient file loaded to the device is correct (if the caller ID signal is distorted, the device won't recognize it).
5. Connect a phone to the analog line of the PBX (instead of to the device's FXO interface) and verify that it displays the caller ID.
6. Configure the following parameters:
   - FXOSeizeLine = 0
   - RTPOnlyMode = 1 (or 2) - the RTP is sent without SIP signaling
   - Coder G.711
   - In the 'Tel to IP Routing' table, route all calls to the PC used for capturing

- EnableCallerID = 0
- RingsBeforeCallerID = 0
- Set the automatic dialing to hotline (e.g., TargetOfChannel7 = 9005,2)
- HotLineToneDuration = 0

The above settings allow the FXO to send RTP without immediately seizing the line after receiving the first ring at your PC.

7. Capture the RTP using Wireshark (you can also use DSP trace) and send the file to AudioCodes.

7.14.7.3 Caller ID on the IP Side

Caller ID is provided by the From header containing the caller's name and "number", for example:

```
From: "David" <SIP:101@10.33.2.2>;tag=35dfsgasd45dg
```

If Caller ID is restricted (received from Tel or configured in the device), the From header is set to:

```
From: "anonymous" <anonymous@anonymous.invalid>; tag=35dfsgasd45dg
```

The P-asserted (or P-preferred) headers are used to present the originating party's caller ID even when the caller ID is restricted. These headers are used together with the Privacy header.

- If Caller ID is restricted:
  - The From header is set to "anonymous" <anonymous@anonymous.invalid>
  - The 'Privacy: id' header is included
  - The P-Asserted-Identity (or P-preferred-Identity) header shows the caller ID
- If Caller ID is allowed:
  - The From header shows the caller ID
  - The 'Privacy: none' header is included
  - The P-Asserted-Identity (or P-preferred-Identity) header shows the caller ID

In addition, the caller ID (and presentation) can be displayed in the Calling Remote-Party-ID header.

The 'Caller Display Information' table (CallerDisplayInfo) is used for the following:

- **FXS interfaces** - to define the caller ID (per port) that is sent to IP.
- **FXO interfaces** - to define the caller ID (per port) that is sent to IP if caller ID isn't detected on the Tel side, or when EnableCallerID = 0.
- **FXS and FXO interfaces** - to determine the presentation of the caller ID (allowed or restricted).
- **To maintain backward compatibility** - when the strings 'Private' or 'Anonymous' are set in the Caller ID/Name field, the caller ID is restricted and the value in the Presentation field is ignored.

The value of the 'Presentation' field that is defined in the 'Caller Display Information' table can be overridden by configuring the 'Presentation' parameter in the 'Tel to IP Source Number Manipulation' table. Therefore, this table can be used to set the presentation for specific calls according to Source / Destination prefixes.

The caller ID can be restricted / allowed (per port) using keypad features KeyCLIR and KeyCLIRDeact (FXS only).

AssertedIdMode defines the header that is used (in the generated INVITE request) to deliver the caller ID (P-Asserted-Identity or P-preferred-Identity). Use the parameter UseTelURIForAssertedID to determine the format of the URI in these headers (sip: or tel:).

EnableRPIheader enables Remote-Party-ID (RPI) headers for calling and called numbers for Tel-to-IP calls.

8 Networking Capabilities

8.1 Ethernet Interface Configuration

The device's Ethernet connection can be configured (using the *ini* file parameter EthernetPhyConfiguration) for one of the following modes:

- **Manual mode:**
  - 10Base-T Half-Duplex or 10Base-T Full-Duplex
  - 100Base-TX Half-Duplex or 100Base-TX Full-Duplex
- **Auto-Negotiation:** chooses common transmission parameters such as speed and duplex mode

Auto-Negotiation falls back to Half-Duplex mode when the opposite port is not in Auto-Negotiation mode, but the speed (i.e., 10/100Base-T) in this mode is always configured correctly. Note that configuring the device to Auto-Negotiation mode while the opposite port is set manually to Full-Duplex (either 10/100Base-T) is invalid (as it causes the device to fall back to Half-Duplex mode while the opposite port is Full-Duplex). It's also invalid to set the device to one of the manual modes when the opposite port is either Auto-Negotiation or not exactly matching (both in speed and in duplex mode).

It's recommended to configure the port for Full-Duplex connections (not Half-Duplex) and 100Base-TX (not 10Base-T) due to the larger bandwidth. It's also strongly recommended to use the same mode in both link partners. Any mismatch configuration can yield unexpected functioning of the Ethernet connection.

Note that when remote configuration is performed, the device should be in the correct Ethernet setting prior to the time this parameter takes effect. When, for example, the device is configured using BootP/TFTP, the device must perform many Ethernet-based transactions prior to reading the *ini* file containing this device configuration parameter. To resolve this problem, the device always uses the last Ethernet setup mode configured. In this way, if you want to configure the device to operate in a new network environment in which the current Ethernet setting of the device is invalid, you should first modify this parameter in the current network so that the new setting holds next time the device is restarted. After reconfiguration has completed, connect the device to the new network and restart it. As a result, the remote configuration process that takes place in the new network uses a valid Ethernet configuration.

8.2 Ethernet Interface Redundancy

The device supports Ethernet redundancy by providing two Ethernet ports, located on the CPU module. The Ethernet port redundancy feature is enabled using the ini file parameter MIIRedundancyEnable. By default, this feature is disabled.

When Ethernet redundancy is implemented, the two Ethernet ports can be connected to the same switch (segment / hub). In this setup, one Ethernet port is active and the other is redundant. If an Ethernet connection failure is detected, the CPU module switches over to the redundant Ethernet port. The CPU issues a Major alarm notifying of the failed physical port. If the first Ethernet port connection is restored, the Major alarm is cleared. The first physical port now becomes the redundant Ethernet port in case of failure with the active physical port (which is currently the second physical port).

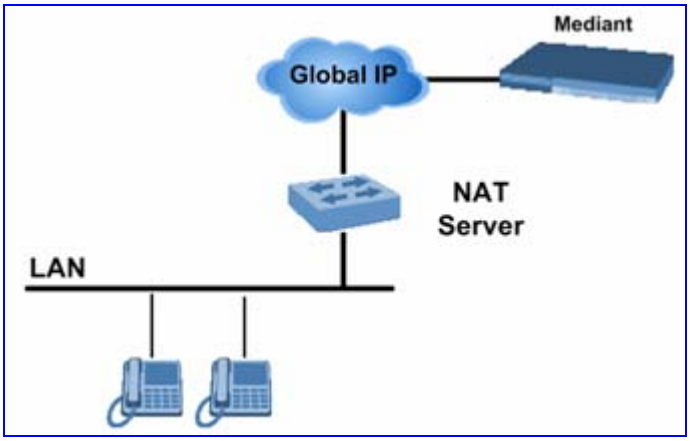
When the CPU module loses all Ethernet connectivity, a Critical alarm is generated:

- When MIIRedundancyEnable is disabled: the alarm is generated when the single physical connection is lost. The alarm is cleared when the single physical connection is restored.
- When MIIRedundancyEnable is enabled: the alarm is generated when both physical connections are lost. The alarm is cleared when one or both of the physical connections are restored.

8.3 NAT (Network Address Translation) Support

Network Address Translation (NAT) is a mechanism that maps a set of internal IP addresses used within a private network to global IP addresses, providing transparent routing to end hosts. The primary advantages of NAT include (1) Reduction in the number of global IP addresses required in a private network (global IP addresses are only used to connect to the Internet); (2) Better network security by hiding its internal architecture.

The following figure illustrates the device's supported NAT architecture.



The design of SIP creates a problem for VoIP traffic to pass through NAT. SIP uses IP addresses and port numbers in its message body and the NAT server can't modify SIP messages and therefore, can't change local to global addresses. Two different streams traverse through NAT: signaling and media. A device (located behind a NAT) that initiates a signaling path has problems in receiving incoming signaling responses (they are blocked by the NAT server). Furthermore, the initiating device must notify the receiving device where to send the media.

To resolve these issues, the following mechanisms are available:

- STUN (refer to "STUN" on page 447)
- First Incoming Packet Mechanism (refer to "First Incoming Packet Mechanism" on page 448)
- RTP No-Op packets according to the avt-rtp-noop draft (refer to "No-Op Packets" on page 448)

For information on SNMP NAT traversal, refer to the *Product Reference Manual*.

8.3.1 STUN

Simple Traversal of UDP through NATs (STUN), based on RFC 3489 is a client / server protocol that solves most of the NAT traversal problems. The STUN server operates in the public Internet and the STUN clients are embedded in end-devices (located behind NAT). STUN is used both for the signaling and the media streams. STUN works with many existing NAT types and does not require any special behavior.

STUN enables the device to discover the presence (and types) of NATs and firewalls located between it and the public Internet. It provides the device with the capability to determine the public IP address and port allocated to it by the NAT. This information is later embedded in outgoing SIP / SDP messages and enables remote SIP user agents to reach the device. It also discovers the binding lifetime of the NAT (the refresh rate necessary to keep NAT 'Pinholes' open).

On startup, the device sends a STUN Binding Request. The information received in the STUN Binding Response (IP address:port) is used for SIP signaling. This information is updated every user-defined period (NATBindingDefaultTimeout).

At the beginning of each call and if STUN is required (i.e., not an internal NAT call), the media ports of the call are mapped. The call is delayed until the STUN Binding Response (that includes a global IP:port) for each media (RTP, RTCP and T.38) is received.

To enable STUN, perform the following:

- Enable the STUN feature using either the Web interface (refer to "Configuring the Application Settings" on page 71) or the *ini* file (set EnableSTUN to 1).
- Define the STUN server address using one of the following methods:
  - Define the IP address of the primary and the secondary (optional) STUN servers using either the Web interface (refer to "Configuring the Application Settings" on page 71) or the *ini* file (STUNServerPrimaryIP and STUNServerSecondaryIP). If the primary STUN server isn't available, the device attempts to communicate with the secondary server.
  - Define the domain name of the STUN server using the *ini* file parameter StunServerDomainName. The STUN client retrieves all STUN servers with an SRV query to resolve this domain name to an IP address and port, sort the server list, and use the servers according to the sorted list.
- Use the *ini* file parameter NATBindingDefaultTimeout to define the default NAT binding lifetime in seconds. STUN is used to refresh the binding information after this time expires.

Notes:

- STUN only applies to UDP (doesn't support TCP and TLS).
- STUN can't be used when the device is located behind a symmetric NAT.
- Use either the STUN server IP address (STUNServerPrimaryIP) or domain name (STUNServerDomainName) method, with priority to the first one.

8.3.2 First Incoming Packet Mechanism

If the remote device resides behind a NAT device, it's possible that the device can activate the RTP/RTCP/T.38 streams to an invalid IP address / UDP port. To avoid such cases, the device automatically compares the source address of the incoming RTP/RTCP/T.38 stream with the IP address and UDP port of the remote device. If the two are not identical, the transmitter modifies the sending address to correspond with the address of the incoming stream. The RTP, RTCP and T.38 can thus have independent destination IP addresses and UDP ports.

You can disable the NAT mechanism by setting the *ini* file parameter DisableNAT to 1. The two parameters EnableIpAddrTranslation and EnableUdpPortTranslation allow you to specify the type of compare operation that occurs on the first incoming packet. To compare only the IP address, set EnableIpAddrTranslation to 1, and EnableUdpPortTranslation to 0. In this case, if the first incoming packet arrives with only a difference in the UDP port, the sending addresses won't change. If both the IP address and UDP port need to be compared, then both parameters need to be set to 1.

8.3.3 No-Op Packets

The device's No-Op packet support can be used to verify Real-Time Transport Protocol (RTP) and T.38 connectivity, and to keep NAT bindings and Firewall pinholes open. The No-Op packets are available for sending in RTP and T.38 formats.

You can control the activation of No-Op packets by using the *ini* file parameter NoOpEnable. If No-Op packet transmission is activated, you can control the time interval in which No-Op packets are sent in the case of silence (i.e., no RTP or T.38 traffic). This is performed using the *ini* file parameter NoOpInterval. For a description of the RTP No-Op *ini* file parameters, refer to "Networking Parameters" on page 296.

- **RTP No-Op:** The RTP No-Op support complies with IETF's draft-wing-avt-rtp-noop-03.txt (titled 'A No-Op Payload Format for RTP'). This IETF document defines a No-Op payload format for RTP. The draft defines the RTP payload type as dynamic. You can control the payload type with which the No-Op packets are sent. This is performed using the RTPNoOpPayloadType *ini* parameter (refer to "Networking Parameters" on page 296). AudioCodes' default payload type is 120.
- **T.38 No-Op:** T.38 No-Op packets are sent only while a T.38 session is activated. Sent packets are a duplication of the previously sent frame (including duplication of the sequence number).

Note: Receipt of No-Op packets is always supported.

8.4 IP Multicasting

The device supports IP Multicasting level 1 according to RFC 2236 (i.e., IGMP version 2) for RTP channels. The device is capable of transmitting and receiving Multicast packets.

8.5 Robust Reception of RTP Streams

This mechanism filters out unwanted RTP streams that are sent to the same port number on the device. These multiple RTP streams can result from traces of previous calls, call control errors, and deliberate attacks. When more than one RTP stream reaches the device on the same port number, the device accepts only one of the RTP streams and rejects the rest of the streams.

The RTP stream is selected according to the following: The first packet arriving on a newly opened channel sets the source IP address and UDP port from which further packets are received. Thus, the source IP address and UDP port identify the currently accepted stream. If a new packet arrives whose source IP address or UDP port are different to the currently accepted RTP stream, one of the following occurs:

- The device reverts to the new RTP stream when the new packet has a source IP address and UDP port that are the same as the remote IP address and UDP port that were stated during the opening of the channel.
- The packet is dropped when the new packet has any other source IP address and UDP port.

8.6 Multiple Routers Support

Multiple routers support is designed to assist the device when it operates in a multiple routers network. The device learns the network topology by responding to Internet Control Message Protocol (ICMP) redirections and caches them as routing rules (with expiration time).

When a set of routers operating within the same subnet serve as devices to that network and intercommunicate using a dynamic routing protocol, the routers can determine the shortest path to a certain destination and signal the remote host the existence of the better route. Using multiple router support, the device can utilize these router messages to change its next hop and establish the best path.

Note: Multiple Routers support is an integral feature that doesn't require configuration.

8.7 Simple Network Time Protocol Support

The Simple Network Time Protocol (SNTP) client functionality generates requests and reacts to the resulting responses using the NTP version 3 protocol definitions (according to RFC 1305). Through these requests and responses, the NTP client synchronizes the system time to a time source within the network, thereby eliminating any potential issues should the local system clock 'drift' during operation. By synchronizing time to a network time source, traffic handling, maintenance, and debugging become simplified for the network administrator.

The NTP client follows a simple process in managing system time: the NTP client requests an NTP update, receives an NTP response, and then updates the local system clock based on a configured NTP server within the network.

The client requests a time update from a specified NTP server at a specified update interval. In most situations, this update interval is every 24 hours based on when the system was restarted. The NTP server identity (as an IP address) and the update interval are user-defined using either the Web interface (refer to "Configuring the Application Settings" on page 71), the *ini* file (NTPServerIP and NTPUpdateInterval respectively), or an SNMP MIB object (refer to the *Product Reference Manual*).

When the client receives a response to its request from the identified NTP server, it must be interpreted based on time zone or location offset that the system is to a standard point of reference called the Universal Time Coordinate (UTC). The time offset that the NTP client uses is configurable using the Web interface (refer to "Configuring the Application Settings" on page 71), the *ini* file (NTPServerUTCOffset), or via an SNMP MIB object (refer to the *Product Reference Manual*).

If required, the clock update is performed by the client as the final step of the update process. The update is performed in such a way as to be transparent to the end users. For instance, the response of the server may indicate that the clock is running too fast on the client. The client slowly robs bits from the clock counter to update the clock to the correct time. If the clock is running too slow, then in an effort to catch the clock up, bits are added to the counter, causing the clock to update quicker and catch up to the correct time. The advantage of this method is that it does not introduce any disparity in the system time that is noticeable to an end user or that could corrupt call timeouts and timestamps.

8.8 IP QoS via Differentiated Services (DiffServ)

DiffServ is an architecture providing different types or levels of service for IP traffic. DiffServ (according to RFC 2474) offers the capability to prioritize certain traffic types depending on their priority, thereby, accomplishing a higher-level QoS at the expense of other traffic types. By prioritizing packets, DiffServ routers can minimize transmission delays for time-sensitive packets such as VoIP packets.

The device can be configured to set a different DiffServ value to IP packets according to their class-of-service: Network, Premium Media, Premium Control, Gold, and Bronze. The DiffServ parameters are described in "Networking Parameters" on page 296.

For the mapping of an application to its class-of-service, refer to "IEEE 802.1p/Q (VLANs and Priority)" on page 451.

8.9 VLANS and Multiple IPs

8.9.1 Multiple IPs

Media, Control, and Management (OAMP) traffic in the device can be assigned one of the following IP addressing schemes:

- **Single IP address for all traffic** (i.e., for Media, Control, and OAMP).
- **Separate IP address for each of the three traffic types:** The different traffic types are separated into three dedicated networks. Instead of a single IP address, the device is assigned three IP addresses and subnet masks, each relating to a different traffic type. This architecture enables you to integrate the device into a three-network environment that is focused on security and segregation. Each entity in the device (e.g., Web and RTP) is mapped to a single traffic type (according to the table in "IEEE 802.1p/Q (VLANs and Priority)" on page 451) in which it operates.

- **Dual IP mode:** The device is assigned two IP addresses for the different traffic types. One IP address is assigned to a combination of two traffic types (Media and Control, OAMP and Control, or OAMP and Media), while the other IP address is assigned to whichever traffic type not included in this combination. For example, a typical scenario using this mode includes one IP address assigned to Control and OAMP, and another IP address assigned to Media.

For detailed information on integrating the device into a VLAN and multiple IPs network, refer to "Getting Started with VLANS and Multiple IPs" on page 453. For detailed information on configuring the multiple IP parameters, refer to "Networking Parameters" on page 296.

Notes:

- A default Gateway is supported only for the Media traffic type; for Control and OAM traffic, use the 'IP Routing' table (refer to "Configuring the IP Routing Table" on page 75).
- The IP address and subnet mask used in the Single IP Network mode are used for the OAM traffic type in the Multiple IP Network mode.

8.9.2 IEEE 802.1p/Q (VLANs and Priority)

The Virtual Local Area Network (VLAN) mechanism enables the device to be integrated into a VLAN-aware environment that includes switches, routers and endpoints. When in VLAN-enabled mode, each packet is tagged with values that specify its priority (class-of-service / IEEE 802.1p) and the identifier (traffic type) of the VLAN to which it belongs (Media, Control, or OAMP / IEEE 802.1Q).

The class-of-service (CoS) mechanism can be utilized to accomplish Ethernet Quality of Service (QoS). Packets sent by the device to the Ethernet network are divided into five different-priority classes (Network, Premium Media, Premium Control, Gold, and Bronze). The priority of each class is determined by a corresponding *ini* file parameter.

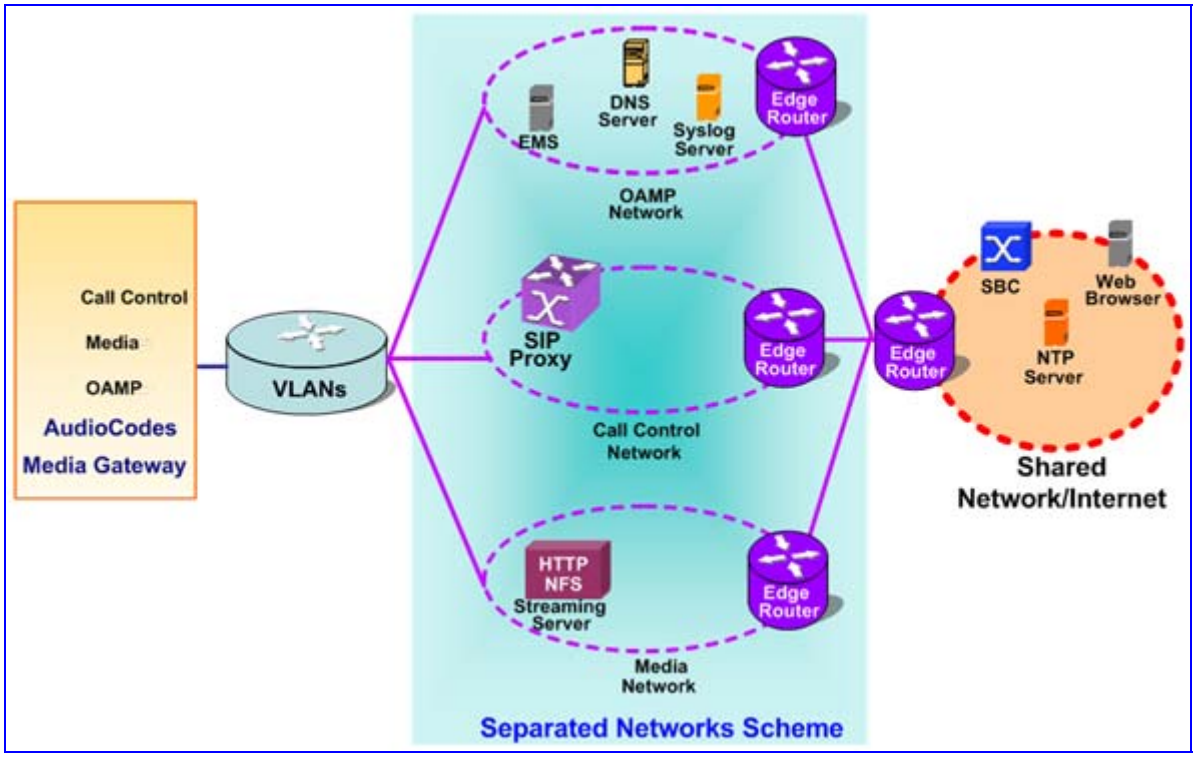
Traffic type tagging can be used to implement Layer 2 VLAN security. By discriminating traffic into separate and independent domains, the information is preserved within the VLAN. Incoming packets received from an incorrect VLAN are discarded.

The traffic tagging mechanism is as follows:

- **Outgoing packets (from the device to the switch):** All outgoing packets are tagged, each according to its interface (Control, Media or OAMP). If the device's native VLAN ID is identical to one of the other IDs (usually to the OAMP's VLAN ID), this ID (e.g., OAMP) is set to zero on outgoing packets (VlanSendNonTaggedOnNative set to 0). This method is called Priority Tagging (p tag without Q tag). If the parameter VlanSendNonTaggedOnNative is set to 1, the device sends regular packets (with no VLAN tag).
- **Incoming packets (from the switch to the device):** The switch sends all packets intended for the device (according to the switch's configuration) to the device without altering them. For packets whose VLAN ID is identical to the switch's PVID, the switch removes the tag and sends a packet. The device accepts only packets that have a VLAN ID identical to one of its interfaces (Control, Media or OAMP). Packets with a VLAN ID that is 0 or untagged packets are accepted only if the device's native VLAN ID is identical to the VLAN ID of one of its interfaces. In this case, the packets are sent to the relevant interface. All other packets are rejected.

Media traffic type is assigned 'Premium media' CoS, Management traffic type is assigned 'Bronze' CoS, and Control traffic type is assigned 'Premium control' CoS. For example, RTP/RTCP traffic is assigned the Media VLAN ID and 'Premium media' CoS, whereas Web traffic is assigned the Management VLAN ID and 'Bronze' CoS. Each of these parameters can be configured with a 802.1p/Q value: traffic type to VLAN ID, and CoS to 802.1p priority.

Figure 8-1: Multiple Network Interfaces and VLANs



Notes:

- For security, the VLAN mechanism is activated only when the device is loaded from the flash memory. Therefore, when using BootP:
  Load an *ini* file with VlanMode set to 1 and SaveConfiguration set to 1. Then (after the device is active) reset the device with TFTP disabled or by using any method except for BootP.
- For information on how to configure VLAN parameters, refer to "Configuring the IP Settings" on page 64.
- The device must be connected to a VLAN-aware switch and the switch's PVID must be equal to the device's native VLAN ID.

The mapping of an application to its CoS and traffic type is shown in the table below:

Table 8-1: Traffic / Network Types and Priority

Application	Traffic / Network Types	Class-of-Service (Priority)
Debugging interface	Management	Bronze
Telnet	Management	Bronze
DHCP	Management	Network

Application	Traffic / Network Types	Class-of-Service (Priority)
Web server (HTTP)	Management	Bronze
SNMP GET/SET	Management	Bronze
Web server (HTTPS)	Management	Bronze
IPSec IKE	Determined by the service	Determined by the service
RTP traffic	Media	Premium media
RTCP traffic	Media	Premium media
T.38 traffic	Media	Premium media
SIP	Control	Premium control
SIP over TLS (SIPS)	Control	Premium control
Syslog	Management	Bronze
ICMP	Management	Determined by the initiator of the request
ARP listener	Determined by the initiator of the request	Network
SNMP Traps	Management	Bronze
DNS client	DNS (EnableDNSasOAM)	Network
NTP	NTP (EnableNTPasOAM)	Depends on traffic type: ▪ Control: Premium control ▪ Management: Bronze
NFS	NFSServers_VlanType in the NFSServers table	Gold

8.9.3 Getting Started with VLANS and Multiple IPs

By default, the device operates without VLANs and multiple IPs, using a single IP address, subnet mask and default Gateway IP address. This section provides an example of the configuration required to integrate the device into a multiple IPs network withVLANs, using the Web interface (refer to "Integrating Using the Web Interface" on page 454) and *ini* file (refer to "Integrating Using the ini File" on page 456). The following table shows an example configuration used in this subsection:

Table 8-2: Example of VLAN and Multiple IPs Configuration

Network Type	IP Address	Subnet Mask	Default Gateway IP Address	VLAN ID	External Routing Rule
OAMP	10.31.174.50	255.255.0.0	0.0.0.0	4	83.4.87.X
Control	10.32.174.50	255.255.0.0	0.0.0.0	5	130.33.4.6
Media	10.33.174.50	255.255.0.0	10.33.0.1	6	--

Notes:

- The values provided in this section are only used as an example.
- Since a default Gateway is available only for the Media network, for the device to be able to communicate with an external device/network on its OAMP and Control networks, IP routing rules must be used.

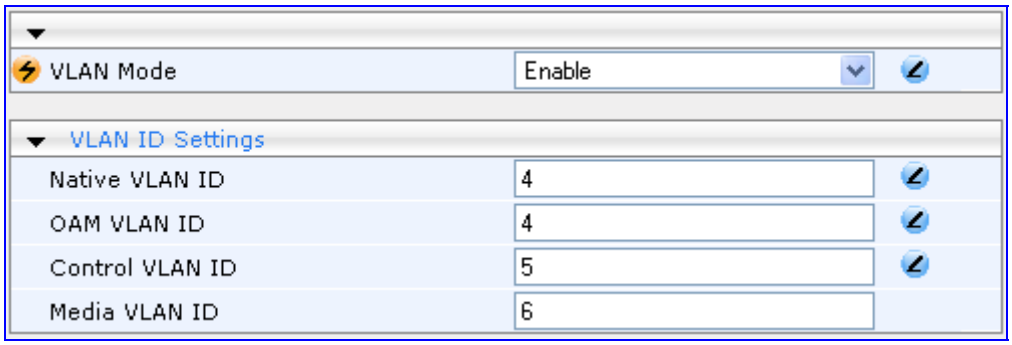
8.9.3.1 Integrating Using the Web Interface

The procedure below describes how to integrate the device into a multiple IPs network withVLANs, using the Web interface.

➢ **To integrate the device into a multiple IPs network withVLANs using the Web interface, take these 6 steps:**

1. Access the Web interface (refer to "Accessing the Web Interface" on page 26).
2. Use the Software Upgrade Wizard (refer to "Software Upgrade Wizard" on page 272) to load and *burn* the firmware version to the device (VLANs and multiple IPs support is available only when the firmware is burned to flash).
3. Configure the VLAN parameters by completing the following steps:
   a. Open the 'IP Settings' page (refer to "Configuring the IP Settings" on page 64).
   b. Modify the VLAN parameters to correspond to the values shown in the following figure:

Figure 8-2: VLAN Configuration in the IP Settings Page

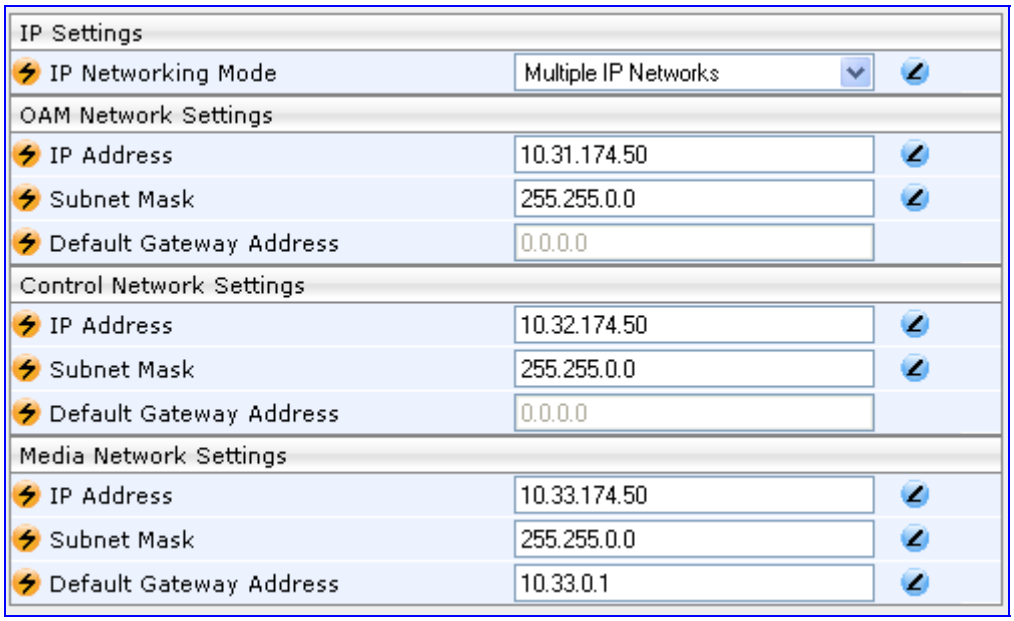


   c. Click the **Submit** button to save your changes.

4. Configure the multiple IP parameters by completing the following steps:

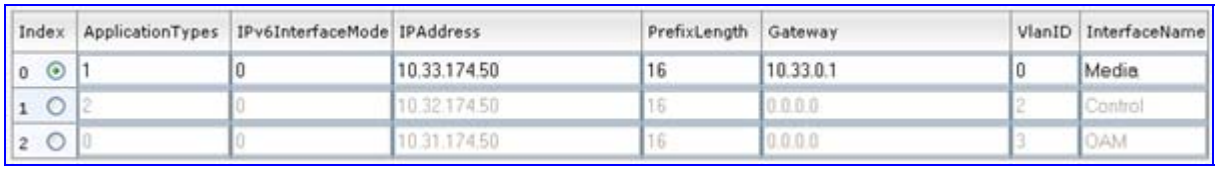
   a. In the 'IP Settings' page, modify the IP parameters to correspond to the values shown in the figure below. Note that the OAM, Control, and Media Network Settings parameters appear only after you select the options 'Multiple IP Networks' or 'Dual IP' in the field 'IP Networking Mode'.

Figure 8-3: OAM, Control, Media IP Configuration in the IP Settings Page



Instead of configuring in the 'IP Settings' page, you can use the 'Multiple Interface Table' page, which is accessed from the 'IP Settings' page by clicking the right-arrow button alongside the label 'Multiple Interface Table' (refer to "Configuring the Multiple Interface Table" on page 68). The 'Multiple Interface Table' page provides greater configuration flexibility whereby you can also assign VLANs to the different interfaces.

Figure 8-4: Multiple Interface Table Page



Index	ApplicationTypes	IPv6InterfaceMode	IPAddress	PrefixLength	Gateway	VlanID	InterfaceName
0	1	0	10.33.174.50	16	10.33.0.1	0	Media
1	2	0	10.32.174.50	16	0.0.0.0	2	Control
2	0	0	10.31.174.50	16	0.0.0.0	3	OAM

   b. Click the **Submit** button to save your changes.

Note: Configure the OAM parameters only if the OAM networking parameters are different from the networking parameters used in the Single IP Network mode.

5. Configure the 'IP Routing' table to define static routing rules for the OAMP and Control networks, since a default gateway isn't supported on these networks:

   a. Open the 'IP Routing Table' page (refer to "Configuring the IP Routing Table" on page 75).

Figure 8-5: Static Routes for OAM/Control in IP Routing Table

   b. Use the **Add New Entry** to add the routing rules listed in the following table:

Destination IP Address	Destination Mask	Gateway IP Address	Hop Count	Interface
87.66.15.8	255.255.255.255	10.13.0.1	20	Control
85.44.115.50	255.255.255.0	10.31.0.1	20	OAMP

6. Save your changes to flash memory (refer to "Saving Configuration" on page 265) and reset the device (refer to "Resetting the Device" on page 262).

8.9.3.2 Integrating Using the ini File

The procedure below describes how to integrate the device into a multiple IPs network with VLANs, using the *ini* file. The procedure below is based on the example setup described in "Getting Started with VLANS and Multiple IPs" on page 453.

➢ To integrate the device into a multiple IPs network withVLANs using the *ini* file, take these 3 steps:

1. Prepare an *ini* file (using the *ini* file table parameter InterfaceTable) with relevant parameters:

   - If the BootP/TFTP utility and the OAMP interface are located on the same network, the Native VLAN ID (VlanNativeVlanId) must be equal to the OAMP VLAN ID (VlanOamVlanId), which in turn must be equal to the PVID of the switch port to which the device is connected. Therefore, set the PVID of the switch port to 4 (in this example).
   - Configure the OAMP parameters only if the OAMP networking parameters are different from the networking parameters used in the Single IP Network mode.
   - The 'IP Routing' table is required to define static routing rules for the OAMP and Control networks since a default Gateway isn't supported for these networks.

Below is an example of an *ini* file containing VLAN and Multiple IPs parameters:

```
; Interface Table Configuration:
[InterfaceTable]
FORMAT InterfaceTable Index = InterfaceTable ApplicationTypes,
InterfaceTable IPv6InterfaceMode, InterfaceTable IPAddress,
InterfaceTable PrefixLength, InterfaceTable Gateway,
InterfaceTable VlanID, InterfaceTable InterfaceName;
InterfaceTable 0 = 0, 0, 10.31.174.50, 16, 0.0.0.0, 4, OAMP;
InterfaceTable 0 = 1, 0, 10.33.174.50, 16, 10.33.0.1, 6, Media;
InterfaceTable 0 = 2, 0, 10.32.174.50, 16, 0.0.0.0, 5, Control;
[\InterfaceTable]

; VLAN related parameters:
VlanMode = 1
VlanNativeVlanId=4

; Routing Table Configuration:
; IP Routing table parameters
RoutingTableDestinationsColumn = 130.33.4.6, 83.4.87.6
RoutingTableDestinationMasksColumn = 255.255.255.255 ,
255.255.255.0
RoutingTableGatewaysColumn = 10.32.0.1 , 10.31.0.1
RoutingTableInterfacesColumn = 2,0
RoutingTableHopsCountColumn = 20,20

; Class Of Service parameters:
VlanNetworkServiceClassPriority = 7
VlanPremiumServiceClassMediaPriority = 6
VlanPremiumServiceClassControlPriority = 6
VlanGoldServiceClassPriority = 4
VlanBronzeServiceClassPriority = 2
NetworkServiceClassDiffServ = 48
PremiumServiceClassMediaDiffServ = 46
PremiumServiceClassControlDiffServ = 40
GoldServiceClassDiffServ = 26
BronzeServiceClassDiffServ = 10

; Application Type for applications:
EnableDNSasOAM = 1
EnableSCTPasControl = 1
EnableNTPasOAM = 1
```

2. Use the BootP/TFTP utility (refer to the *Product Reference Manual*) to load and *burn* the firmware version and the *ini* file you prepared in the previous step to the device (multiple IPs and VLANs support is available only when the firmware is burned to flash).
3. Reset the device after disabling it on the BootP/TFTP utility.

Instead of using the *ini* file table parameter InterfaceTable, you can configure multiple IPs and VLANs using the individual *ini* file parameters, as shown below:

```
; VLAN Configuration
VlanMode=1
VlanOamVlanId=4
VlanNativeVlanId=4
VlanControlVlanId=5
VlanMediaVlanID=6

; Multiple IPs Configuration
EnableMultipleIPs=1
LocalMediaIPAddress=10.33.174.50
LocalMediaSubnetMask=255.255.0.0
LocalMediaDefaultGW=10.33.0.1
LocalControlIPAddress=10.32.174.50
LocalControlSubnetMask=255.255.0.0
LocalControlDefaultGW=0.0.0.0
LocalOAMPAddress=10.31.174.50
LocalOAMSubnetMask=255.255.0.0
LocalOAMDefaultGW=0.0.0.0

; IP Routing table parameters
RoutingTableDestinationsColumn = 130.33.4.6, 83.4.87.6
RoutingTableDestinationMasksColumn = 255.255.255.255,
255.255.255.0
RoutingTableGatewaysColumn = 10.32.0.1 , 10.31.0.1
RoutingTableInterfacesColumn = 1 , 0
RoutingTableHopsCountColumn = 20,20
```

9 Advanced PSTN Configuration

9.1 Clock Settings

The device's clock settings can be configured to either generate its own timing signals, use an internal clock, or recover them from one of the E1/T1 trunks.

➢ To use the device's internal clock source, configure the following parameters:

- TDMBusClockSource = 1
- ClockMaster = 1 (for all trunks)

➢ To use the recovered clock option, configure the following parameters:

- TDMBusClockSource = 4
- ClockMaster_x = 0 (for all 'slave' trunks connected to PBX#1)
- ClockMaster_x = 1 (for all 'master' trunks connected to PBX#2)

The above assumes that the device recovers its internal clock from one of the 'slave' trunks connected to PBX#1 and provides clock to PBX#2 on its 'master' trunks.

In addition, it's necessary to define from which of the 'slave' trunks the device recovers its clock:

- TDMBusPSTNAutoClockEnable = 1 (device automatically selects one of the connected 'slave' trunks)
  - Or -
- TDMBusLocalReference = # (Trunk index: 0 to 3, default = 0)

Notes:

- To configure the TDM Bus Clock Source parameters using the Web interface, refer to "Configuring the TDM Bus Settings" on page 251.
- When the device is used in a 'non-span' configuration, the internal device clock must be used (as explained above).

9.2 Release Reason Mapping

This section describes the available mapping mechanisms of SIP Responses to Q.850 Release Causes and vice versa. The existing mapping of ISDN Release Causes to SIP Responses is described in "Fixed Mapping of ISDN Release Reason to SIP Response" on page 460 and "Fixed Mapping of SIP Response to ISDN Release Reason" on page 462. To override this hard-coded mapping and flexibly map SIP Responses to ISDN Release Causes, use the *ini* file (CauseMapISDN2SIP and CauseMapSIP2ISDN, as described in "ISDN and CAS Interworking-Related Parameters" on page 348) or the Web interface (refer to "Release Cause Mapping" on page 211).

It is also possible to map the less commonly used SIP Responses to a single default ISDN Release Cause. Use the parameter DefaultCauseMapISDN2IP (described in "ISDN and CAS Interworking-Related Parameters" on page 348) to define a default ISDN Cause that is always used except when the following Release Causes are received: Normal Call Clearing (16), User Busy (17), No User Responding (18) or No Answer from User (19). This mechanism is only available for Tel-to-IP calls.

9.2.1 Reason Header

The device supports the Reason header according to RFC 3326. The Reason header conveys information describing the disconnection cause of a call:

- **Sending Reason header:** If a call is disconnected from the Tel side (ISDN), the Reason header is set to the received Q.850 cause in the appropriate message (BYE / CANCEL / final failure response) and sent to the SIP side. If the call is disconnected because of a SIP reason, the Reason header is set to the appropriate SIP response.
- **Receiving Reason header:** If a call is disconnected from the IP side and the SIP message includes the Reason header, it is sent to the Tel side according to the following logic:
  - If the Reason header includes a Q.850 cause, it is sent as is.
  - If the Reason header includes a SIP response:
    - If the message is a final response, the response status code is translated to Q.850 format and passed to ISDN.
    - If the message isn't a final response, it is translated to a Q.850 cause.
  - When the Reason header is received twice (i.e., SIP Reason and Q.850), the Q.850 takes precedence over the SIP reason and is sent to the Tel side.

9.2.2 Fixed Mapping of ISDN Release Reason to SIP Response

The following table describes the mapping of ISDN release reason to SIP response.

Table 9-1: Mapping of ISDN Release Reason to SIP Response

ISDN Release Reason	Description	SIP Response	Description
1	Unallocated number	404	Not found
2	No route to network	404	Not found
3	No route to destination	404	Not found
6	Channel unacceptable	406*	Not acceptable
7	Call awarded and being delivered in an established channel	500	Server internal error
16	Normal call clearing	-*	BYE
17	User busy	486	Busy here
18	No user responding	408	Request timeout
19	No answer from the user	480	Temporarily unavailable
21	Call rejected	403	Forbidden
22	Number changed w/o diagnostic	410	Gone

ISDN Release Reason	Description	SIP Response	Description
26	Non-selected user clearing	404	Not found
27	Destination out of order	502	Bad gateway
28	Address incomplete	484	Address incomplete
29	Facility rejected	501	Not implemented
30	Response to status enquiry	501*	Not implemented
31	Normal unspecified	480	Temporarily unavailable
34	No circuit available	503	Service unavailable
38	Network out of order	503	Service unavailable
41	Temporary failure	503	Service unavailable
42	Switching equipment congestion	503	Service unavailable
43	Access information discarded	502*	Bad gateway
44	Requested channel not available	503*	Service unavailable
47	Resource unavailable	503	Service unavailable
49	QoS unavailable	503*	Service unavailable
50	Facility not subscribed	503*	Service unavailable
55	Incoming calls barred within CUG	403	Forbidden
57	Bearer capability not authorized	403	Forbidden
58	Bearer capability not presently available	503	Service unavailable
63	Service/option not available	503*	Service unavailable
65	Bearer capability not implemented	501	Not implemented
66	Channel type not implemented	480*	Temporarily unavailable
69	Requested facility not implemented	503*	Service unavailable
70	Only restricted digital information bearer capability is available	503*	Service unavailable
79	Service or option not implemented	501	Not implemented
81	Invalid call reference value	502*	Bad gateway
82	Identified channel does not exist	502*	Bad gateway
83	Suspended call exists, but this call identity does not	503*	Service unavailable
84	Call identity in use	503*	Service unavailable
85	No call suspended	503*	Service unavailable
86	Call having the requested call identity has been cleared	408*	Request timeout
87	User not member of CUG	503	Service unavailable
88	Incompatible destination	503	Service unavailable
91	Invalid transit network selection	502*	Bad gateway

ISDN Release Reason	Description	SIP Response	Description
95	Invalid message	503	Service unavailable
96	Mandatory information element is missing	409*	Conflict
97	Message type non-existent or not implemented	480*	Temporarily not available
98	Message not compatible with call state or message type non-existent or not implemented	409*	Conflict
99	Information element non-existent or not implemented	480*	Not found
100	Invalid information elements contents	501*	Not implemented
101	Message not compatible with call state	503*	Service unavailable
102	Recovery of timer expiry	408	Request timeout
111	Protocol error	500	Server internal error
127	Interworking unspecified	500	Server internal error

* Messages and responses were created because the 'ISUP to SIP Mapping' draft doesn't specify their cause code mapping.

9.2.3 Fixed Mapping of SIP Response to ISDN Release Reason

The following table describes the mapping of SIP response to ISDN release reason.

Table 9-2: Mapping of SIP Response to ISDN Release Reason

SIP Response	Description	ISDN Release Reason	Description
400*	Bad request	31	Normal, unspecified
401	Unauthorized	21	Call rejected
402	Payment required	21	Call rejected
403	Forbidden	21	Call rejected
404	Not found	1	Unallocated number
405	Method not allowed	63	Service/option unavailable
406	Not acceptable	79	Service/option not implemented
407	Proxy authentication required	21	Call rejected
408	Request timeout	102	Recovery on timer expiry
409	Conflict	41	Temporary failure
410	Gone	22	Number changed w/o diagnostic
411	Length required	127	Interworking
413	Request entity too long	127	Interworking
414	Request URI too long	127	Interworking

SIP Response	Description	ISDN Release Reason	Description
415	Unsupported media type	79	Service/option not implemented
420	Bad extension	127	Interworking
480	Temporarily unavailable	18	No user responding
481*	Call leg/transaction doesn't exist	127	Interworking
482*	Loop detected	127	Interworking
483	Too many hops	127	Interworking
484	Address incomplete	28	Invalid number format
485	Ambiguous	1	Unallocated number
486	Busy here	17	User busy
488	Not acceptable here	31	Normal, unspecified
500	Server internal error	41	Temporary failure
501	Not implemented	38	Network out of order
502	Bad gateway	38	Network out of order
503	Service unavailable	41	Temporary failure
504	Server timeout	102	Recovery on timer expiry
505*	Version not supported	127	Interworking
600	Busy everywhere	17	User busy
603	Decline	21	Call rejected
604	Does not exist anywhere	1	Unallocated number
606*	Not acceptable	38	Network out of order

***** Messages and responses were created because the 'ISUP to SIP Mapping' draft doesn't specify their cause code mapping.

9.3 ISDN Overlap Dialing

Overlap dialing is a dialing scheme used by several ISDN variants to send and / or receive called number digits one after the other (or several at a time). This is in contrast to en-bloc dialing in which a complete number is sent.

The device can optionally support ISDN overlap dialing for incoming ISDN calls for the entire device by setting the *ini* file parameter ISDNRxOverlap to 1, or per E1/T1 span by setting ISDNRxOverlap_x to 1 (where *x* represents the number of the trunk). For configuring ISDN overlap dialing using the Web interface, refer to "Configuring the Trunk Settings" on page 96.

To play a Dial tone to the ISDN user side when an empty called number is received, set ISDNINCallsBehavior = 65536 (bit #16). This results in the Progress Indicator to be included in the SetupAck ISDN message.

The device stops collecting digits (for ISDN-to-IP calls) when:

- The sending device transmits a 'sending complete' IE in the ISDN Setup or the following INFO messages to signal that no more digits are going to be sent.
- The inter-digit timeout (configured by the parameter TimeBetweenDigits) expires. The default for this timeout is 4 seconds.
- The maximum allowed number of digits (configured by the parameter MaxDigits) is reached. The default is 30 digits.
- A match is found with the defined digit map (configured by the parameter, DigitMapping).

Relevant parameters (described in "PSTN Parameters" on page 344):

- ISDNRxOverlap
- ISDNRxOverlap_x
- TimeBetweenDigits
- MaxDigits
- ISDNInCallsBehavior
- DigitMapping

9.4 Using ISDN NFAS

In regular T1 ISDN trunks, a single 64 kbps channel carries signaling for the other 23 B-channels of that particular T1 trunk. This channel is called the D-channel and usually resides on timeslot # 24.

The ISDN Non-Facility Associated Signaling (NFAS) feature enables use of a single D-channel to control multiple PRI interfaces.

With NFAS it is possible to define a group of T1 trunks, called an NFAS group, in which a single D-channel carries ISDN signaling messages for the entire group. The NFAS group's B-channels are used to carry traffic such as voice or data. The NFAS mechanism also enables definition of a backup D-channel on a different T1 trunk, to be used if the primary D-channel fails.

The NFAS group comprises several T1 trunks. Each T1 trunk is called an 'NFAS member'. The T1 trunk whose D-channel is used for signaling is called the 'Primary NFAS Trunk'. The T1 trunk whose D-channel is used for backup signaling is called the 'Backup NFAS Trunk'. The primary and backup trunks each carry 23 B-channels while all other NFAS trunks each carry 24 B-channels.

The device supports multiple NFAS groups. Each group should contain different T1 trunks.

The NFAS group is identified by an NFAS GroupID number (possible values are 1, 2, 3 and 4). To assign a number of T1 trunks to the same NFAS group, use the *ini* file parameter NFASGroupNumber_x = groupID (where *x* is the physical trunkID -- 0 to 3) or the Web interfacer (refer to "Configuring the Trunk Settings" on page 96).

The parameter 'DchConfig_x = Trunk_type' defines the type of NFAS trunk. Trunk_type is set to 0 for the primary trunk, to 1 for the backup trunk, and to 2 for an ordinary NFAS trunk. 'x' depicts the physical trunkID (0 to 3). You can also use the Web interface (refer to "Configuring the Trunk Settings" on page 96).

For example, to assign the first four T1 trunks to NFAS group #1, in which trunk #0 is the primary trunk and trunk #1 is the backup trunk, use the following configuration:

```
NFASGroupNumber 0 = 1
NFASGroupNumber 1 = 1
NFASGroupNumber 2 = 1
NFASGroupNumber 3 = 1
DchConfig 0 = 0              ;Primary T1 trunk
DchConfig 1 = 1              ;Backup T1 trunk
DchConfig 2 = 2              ;24 B-channel NFAS trunk
DchConfig 3 = 2              ;24 B-channel NFAS trunk
```

The NFAS parameters are described in "PSTN Parameters" on page 344.

9.4.1 NFAS Interface ID

Several ISDN switches require an additional configuration parameter per T1 trunk that is called 'Interface Identifier'. In NFAS T1 trunks, the Interface Identifier is sent explicitly in Q.931 Setup / Channel Identification IE for all NFAS trunks, except for the B-channels of the Primary trunk (refer to note below).

The Interface ID can be defined per member (T1 trunk) of the NFAS group, and must be coordinated with the configuration of the Switch. The default value of the Interface ID is identical to the number of the physical T1 trunk (0 for the first trunk, 1 for the second T1 trunk, and so on, up to 3).

To define an explicit Interface ID for a T1 trunk (that is different from the default), use the following parameters:

- ISDNIBehavior_x = 512 (x = 0 to 3 identifying the device's physical trunk)
- ISDNNFASInterfaceID_x = ID (x = 0 to 255)

Notes:

- Usually the Interface Identifier is included in the Q.931 Setup/Channel Identification IE only on T1 trunks that doesn't contain the D-channel. Calls initiated on B-channels of the Primary T1 trunk, by default, don't contain the Interface Identifier. Setting the parameter ISDNIBehavior_x to 2048' forces the inclusion of the Channel Identifier parameter also for the Primary trunk.
- The parameter ISDNNFASInterfaceID_x = ID can define the 'Interface ID' for any Primary T1 trunk, even if the T1 trunk is not a part of an NFAS group. However, to include the Interface Identifier in Q.931 Setup/Channel Identification IE configure ISDNIBehavior_x = 2048 in the *ini* file.

9.4.2 Working with DMS-100 Switches

The DMS-100 switch requires the following NFAS Interface ID definitions:

- InterfaceID #0 for the Primary trunk
- InterfaceID #1 for the Backup trunk
- InterfaceID #2 for a 24 B-channel T1 trunk
- InterfaceID #3 for a 24 B-channel T1 trunk, and so on for subsequent T1 trunks

For example, if four T1 trunks on a device are configured as a single NFAS group with Primary and Backup T1 trunks that is used with a DMS-100 switch, the following parameters should be used:

```
NFASGroupNumber_0 = 1
NFASGroupNumber_1 = 1
NFASGroupNumber_2 = 1
NFASGroupNumber_3 = 1
DchConfig_0 = 0    ;Primary T1 trunk
DchConfig_1 = 1    ;Backup T1 trunk
DchConfig_2 = 2    ;B-Channel NFAS trunk
DchConfig_3 = 2    ;B-channel NFAS trunk
```

If there is no NFAS Backup trunk, the following configuration should be used:

```
ISDNNFASInterfaceID_0 = 0
ISDNNFASInterfaceID_1 = 2
ISDNNFASInterfaceID_2 = 3
ISDNNFASInterfaceID_3 = 4
ISDNIBehavior = 512   ;This parameter should be added because of
;ISDNNFASInterfaceID coniguration above
NFASGroupNumber_0 = 1
NFASGroupNumber_1 = 1
NFASGroupNumber_2 = 1
NFASGroupNumber_3 = 1
DchConfig_0 = 0    ;Primary T1 trunk
DchConfig_1 = 2    ;B-Channel NFAS trunk
DchConfig_2 = 2    ;B-Channel NFAS trunk
DchConfig_3 = 2    ;B-channel NFAS trunk
```

9.4.3 Creating an NFAS-Related Trunk Configuration On-The-Fly

The procedures for creating and deleting an NFAS group on-the-fly must be performed in the correct order, as described below.

➢ To create an NFAS Group, take these 3 steps:

1. If there's a backup ('secondary') trunk for this group, it must be configured first.
2. Configure the primary trunk before configuring any NFAS ('slave') trunk.
3. Configure NFAS ('slave') trunks.

➢ To stop / delete an NFAS Group, take these 3 steps:

1. Stop / delete all NFAS ('slave') trunks.
2. Stop / delete the backup trunk if a backup trunk exists.
3. Stop / delete the primary trunk.

Notes:

- All trunks in the group must be configured with the same values for trunk parameters TerminationSide, ProtocolType, FramingMethod, and LineCode.
- After stopping or deleting the backup trunk, delete the group and then reconfigure it.

9.5 Redirect Number and Calling Name (Display)

The following tables define the device's redirect number and calling name (Display) support for various PRI variants according to NT (Network Termination) / TE (Termination Equipment) interface direction:

Table 9-3: Calling Name (Display)

NT/TE Interface	DMS-100	NI-2	4/5ESS	Euro ISDN	QSIG
NT-to-TE	Yes	Yes	Yes	Yes	Yes
TE-to-NT	Yes	Yes	Yes	No	Yes

Table 9-4: Redirect Number

NT/TE Interface	DMS-100	NI-2	4/5ESS	Euro ISDN	QSIG
NT-to-TE	Yes	Yes	Yes	Yes	Yes
TE-to-NT	Yes	Yes	Yes	Yes*	Yes

* When using ETSI DivertingLegInformation2 in a Facility IE (not Redirecting Number IE).

10 Media Server Capabilities

This section provides information on the device's media server capabilities:

- Multi-party conferencing (refer to "Conference Server" on page 469)
- Playing and recording Announcements (refer to "Announcement Server" on page 483)
- IP-to-IP Transcoding (refer to "IP-to-IP Transcoding" on page 494)

Note: This section is applicable only to Mediant 1000.

The device conference, transcoding, announcement and media server applications can be used separately, each on a different platform, or all on the same device. The SIP URI name in the INVITE message is used to identify the resource (media server, conference or announcement) to which the SIP session is addressed.

The number of DSP channels that are allocated for IP conferences, transcoding and IP announcements is determined by the parameter MediaChannels. Other DSP channels can be used for PSTN media server.

The device SIP implementation is based on the decomposition model described in the following IETF drafts:

- 'A Multi-party Application Framework for SIP' (draft-ietf-sipping-cc-framework-06.txt)
- 'Models for Multi Party Conferencing in SIP' (draft-ietf-sipping-conferencing-framework-05.txt)
- 'A Framework for Conferencing with the Session Initiation Protocol (SIP)' (RFC 4353)
- 'Basic Network Media Services with SIP' (RFC 4240)
- 'Media Server Control Markup Language (MSCML) and Protocol' (draft-vandyke-mscml-06.txt)

Note: To use the device's advanced Announcement capabilities, it's essential that the *ini* file parameter AMSProfile be set to 1.

10.1 Conference Server

The device supports dial-in, multi-party conferencing. In conference applications, the device functions as a centralized conference bridge. In ad-hoc or prearranged conferences, users 'invite' the conference bridge. The conference bridge mixes the media and sends it to all participants.

The device supports the following interfaces for conferencing:

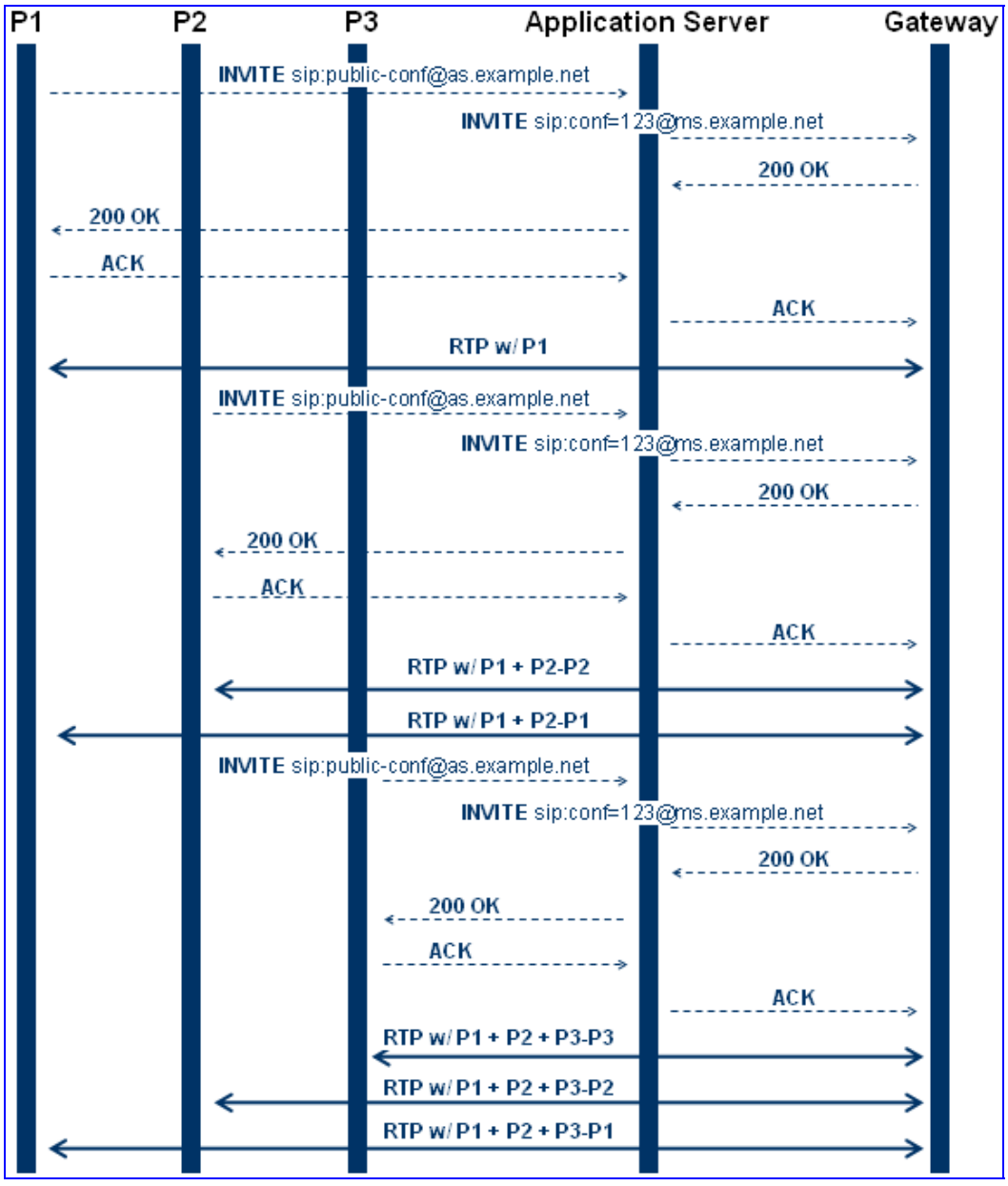
- Simple, according to NetAnn (refer to "Simple Conferencing (NetAnn)" on page 470)
- Advanced, according to MSCML (refer to "Advanced Conferencing (MSCML)" on page 472)

Note: The conference application is a special order option.

10.1.1 Simple Conferencing (NetAnn)

10.1.1.1 SIP Call Flow

Figure 10-1: Simple Conferencing SIP Call Flow



10.1.1.2 Creating a Conference

The device creates a conference call when the first user joins the conference. To create a conference, the Application Server sends a regular SIP INVITE message to the device. The User Part of that Request-URI includes both the Conference Service Identifier (indicating that the requested Media Service is a Conference) and a Unique Conference Identifier (identifying a specific instance of a conference).

```
INVITE sip: conf100@audiocodes.com SIP/2.0
```

By default, a request to create a conference reserves three resources on the device. It is possible to reserve a larger number of resources in advance by adding the number of required participants to the User Part of the Request-URI. For example, '6conf100' reserves six resources for the duration of the conference. If the device can allocate the requested number of resources, it responds with a 200 OK.

The Conference Service Identifier can be set using the *ini* file (ConferenceID) or Web interface (refer to "Supplementary Services" on page 176). By default, it is set to 'conf'.

10.1.1.3 Joining a Conference

To join an existing conference, the Application Server sends a SIP INVITE message with the same Request-URI as the one that created the conference. Each conference participant can use a different coder negotiated with the device using usual SIP negotiation.

If more than the initially requested number of participants try to join the conference (i.e., four resources were reserved and a fifth INVITE is received) and the device has an available resource, the request is granted.

If an INVITE to join an existing conference is received with a request to reserve a larger number of participants than initially requested, it is granted if the device has available resources. A request for a smaller number of participants is not granted as this may create a situation where existing legs would need to be disconnected.

The maximal number of participants in a single conference is 60. The maximal number of participants that actually participate in the mix at a given time is three (the loudest legs).

The Application Server can place a participant on Hold/Un-hold by sending the appropriate SIP Re-INVITE on that participant dialog.

10.1.1.4 Terminating a Conference

The device never disconnects an existing conference leg. If a BYE is received on an existing leg, it is disconnected, but the resource is still saved if the same leg (or a different one) wants to re-join the conference. This logic occurs only for the initial number of reserved legs.

For example:

1. INVITE reserves three legs.
2. A, B, and C join the conference.
3. A disconnects.
4. A joins (guaranteed).
5. D joins.
6. A disconnects.
7. A joins (not guaranteed).

Sending a BYE request to the device terminates the participant's SIP session and removes it from the conference. The final BYE from the last participant ends the conference and releases all conference resources.

10.1.1.5 PSTN Participants

Adding PSTN participants is done by performing a loopback from the IP side (the device's IP address is configured in the 'Tel to IP Routing' table). If the destination phone number in the incoming call from the PSTN is equal to the Conference Service Identifier and Unique Conference Identifier, the participant joins the conference.

A PSTN participant uses two DSP channels (caused by the IP loopback).

10.1.2 Advanced Conferencing (MSCML)

10.1.2.1 Creating a Conference

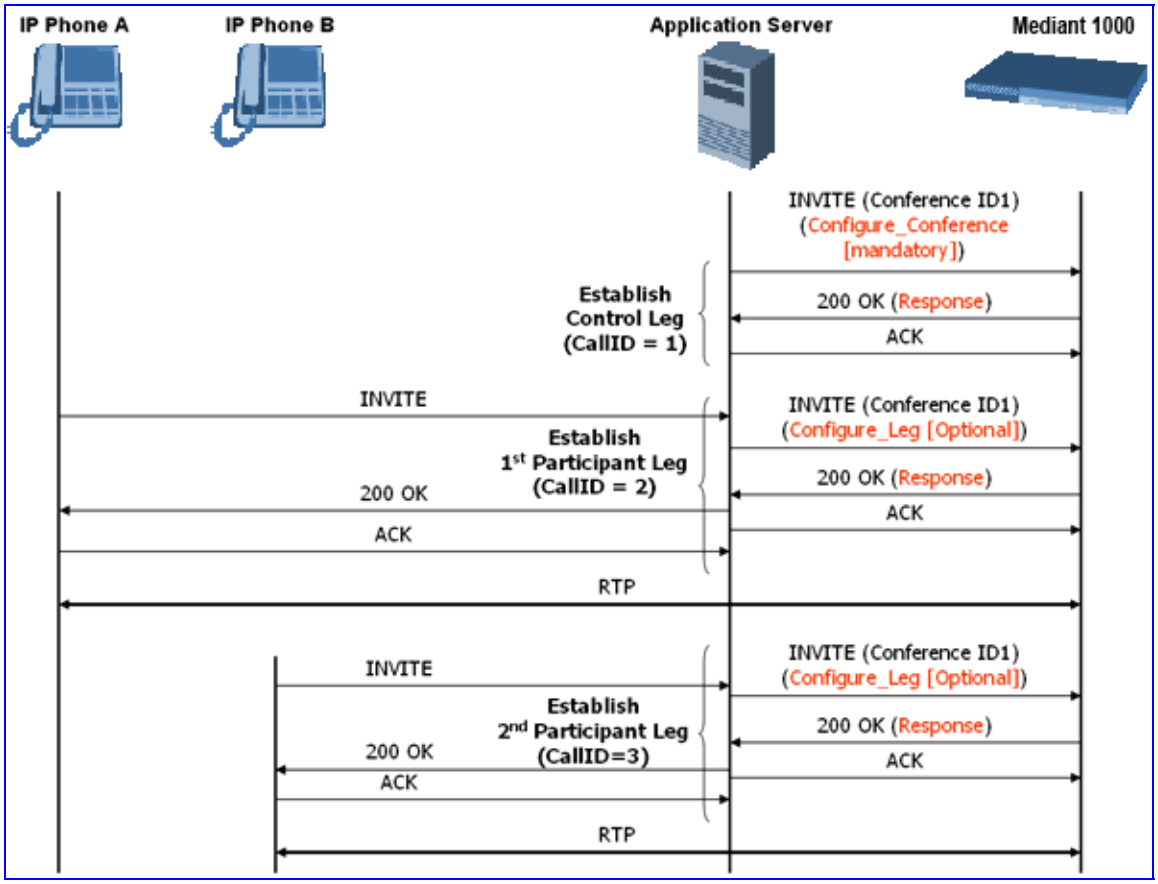
The device creates a conference call when the first INVITE is received from the Application Server (same as NetAnn). The Unique Conference Identifier is used to join participants to the same conference. This first INVITE must include a <configure_conference> MSCML request body. If this body is not included, a simple conference is established. This first leg is the Control Leg, which is different from a regular Participant Leg. The Control Leg is used to perform operations for the whole conference.

The MSCML response to the first INVITE is sent in the 200 OK SIP response. If no error occurrs, the response is:
<response request="configure_conference" code="200" text="OK"/>.

The <configure_conference> can include the following attributes:

- **Id:** identification number of the MSCML request. This is used to correlate between MSCML requests and responses.
- **Reservedtalkers:** defines the maximum number of talker legs. As the device does not support “listener only” legs, this actually sets the maximum number of participants in the conference. The device reserves this number of participants for the entire duration of the conference. If a participant leg decides to leave the conference by issuing a BYE, the resource is not freed, thereby allowing that same leg (or a new one) to join at any stage.
- **Reserveconfmedia:** determines if Media Services such as Play or Record can be applied to the conference. If set to Yes, the device reserves the necessary amount of resources to play an announcement to the whole conference or record the whole conference. The Application Server can change the value of reserveconfmedia during an existing conference. By default, reserveconfmedia is set to Yes.

Figure 10-2: Advanced Conferencing SIP Call Flow



10.1.2.2 Joining a Conference

To join an existing conference, the Application Server sends a SIP INVITE message with the same Request-URI as the one that created the conference. The INVITE message may include a <configure_leg> MSCML request body. If not included, defaults are used for that leg attributes.

The <configure_leg> can include the following attributes:

- **Id:** identification number of the MSCML request. This is used to correlate between MSCML requests and responses.
- **Type:** Talker / Listener. If set to Listener, the incoming RTP from that leg does not participate in the conference mix. The default is Talker.
- **Mixmode:**
  - Full: RTP from this leg participates in the mix (default).
  - Mute: RTP from this leg is not participating in the mix.

10.1.2.3 Modifying a Conference

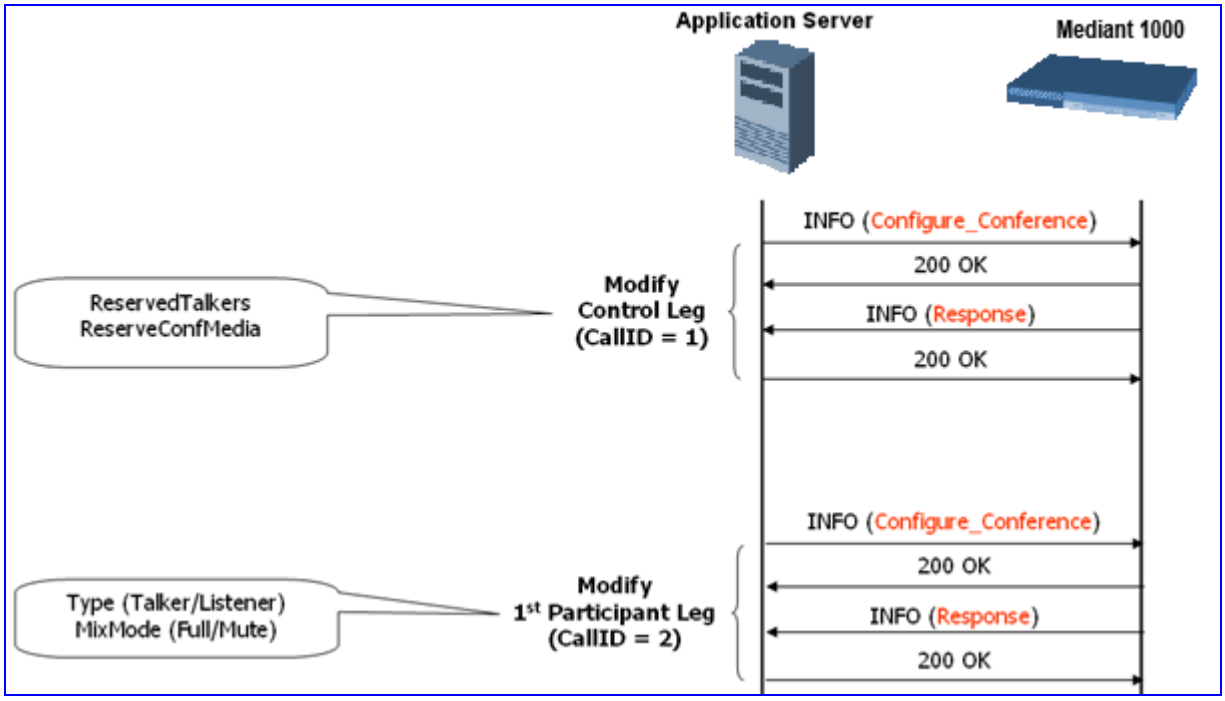
To modify an existing conference, INFO messages are used. Each INFO message carries an MSCML request. The MSCML response is included in an INFO message back from the device to the Application Server. It is possible to modify an entire conference (by issuing requests on the Control Leg) or only a certain participant (by issuing requests on that specific leg).

To modify the entire conference, a <configure_conference> MSCML request body is sent in an INFO message on the Control Leg SIP dialog. Using this request, the Application Server can modify the following attributes:

- **Reservedtalkers:** If the Application Server sets a number that is lower than the initial number requested in the INVITE, then the request is not granted. If the number is higher than the initial number, the device sends a success response in the response INFO.
- **Reserveconfmedia:** If the necessary resources for applying Media Services on the entire conference were reserved in advance, then by setting reserveconfmedia to Yes, it is reserved. If set to No, the device can free the resource.

To modify a certain Participant Leg, a <configure_leg> MSCML request body is sent in an INFO message on that leg SIP dialog. Using this request, the Application Server can modify any of the attributes defined for the <configure_leg> request.

Figure 10-3: Modifying a Conference - SIP Call Flow

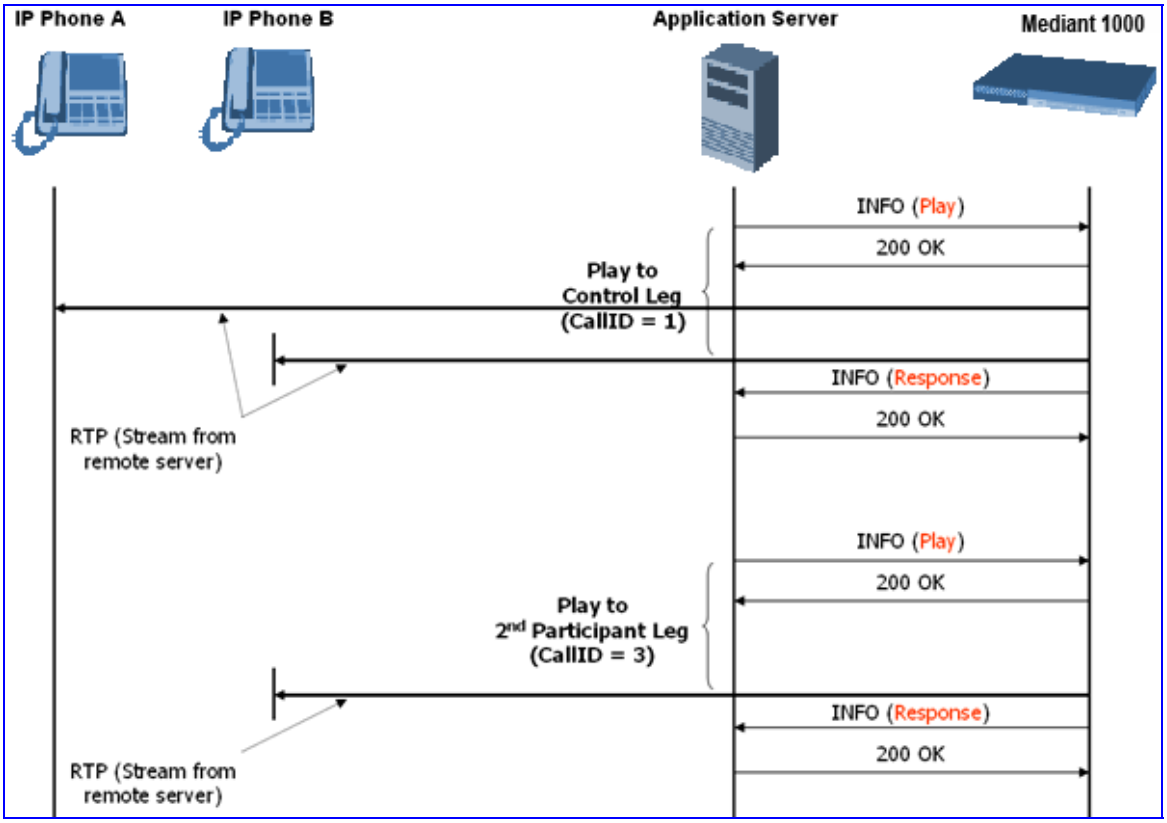


10.1.2.4 Applying Media Services on a Conference

The Application Server can issue a Media Service request (<play>, <playcollect>, or <playrecord>) on either the Control Leg or a specific Participant Leg. For a Participant Leg, all three requests are applicable. For the Control Leg, the <playcollect> is not applicable as there is no way to collect digits from the whole conference.

When issuing a Media Service on the Control Leg, it affects all Participant Legs in the conference, e.g., play an announcement. When issuing a Media Service on a Participant Leg, it affects the specific leg only.

Figure 10-4: Applying Media Services on a Conference -- SIP Call Flow



10.1.2.5 Active Speaker Notification

After an advanced conference is established, the Application Server can subscribe to the device to receive notifications of the current set of active speakers in a conference at any given moment. This feature is referred to as *Active Speaker Notification* (ASN) and is designed according to the MSCML standard. Notifications provide information on the number of active participants and their details.

The notifications are sent unsolicited at specific intervals requested by the application and only when a change in the number of active conference speakers occurs. If a change in the speakers list occurs, the server issues an INVITE to the specific SIP UA, and then transfers the call to the UA.

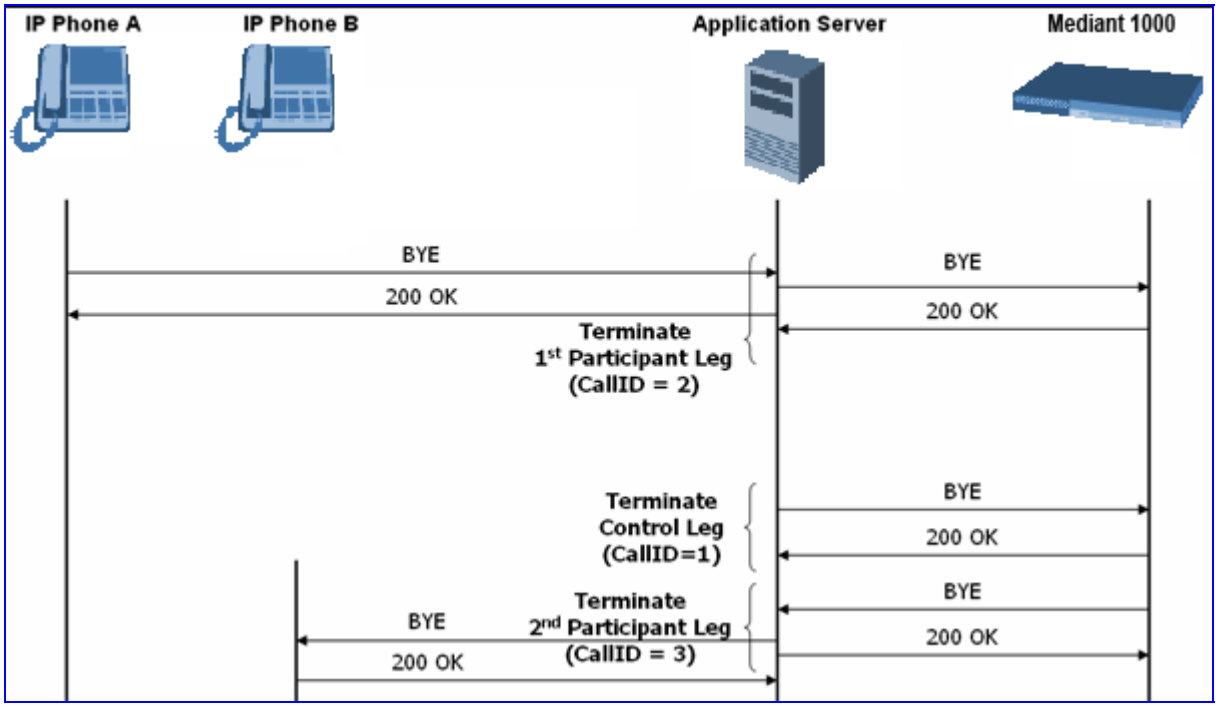
Event notifications are sent in SIP INFO messages, as shown in the example below of XML Response Generated for ASN:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
<notification>
<conference uniqueID="3331" numtalkers="1">
<activetalkers>
<talker callID="9814266171512000193619@10.8.27.118"/>
</activetalkers>
</conference>
</notification>
</MediaServerControl>
```

10.1.2.6 Terminating a Conference

To remove a leg from a conference, the Application Server issues a SIP BYE request on the selected dialog representing the conference leg. The Application Server can terminate all legs in a conference by issuing a SIP BYE request on the Control Leg. If one or more participants are still in the conference when the device receives a SIP BYE request on the Control Leg, the device issues SIP BYE requests on all of the remaining conference legs to ensure a clean up of the legs.

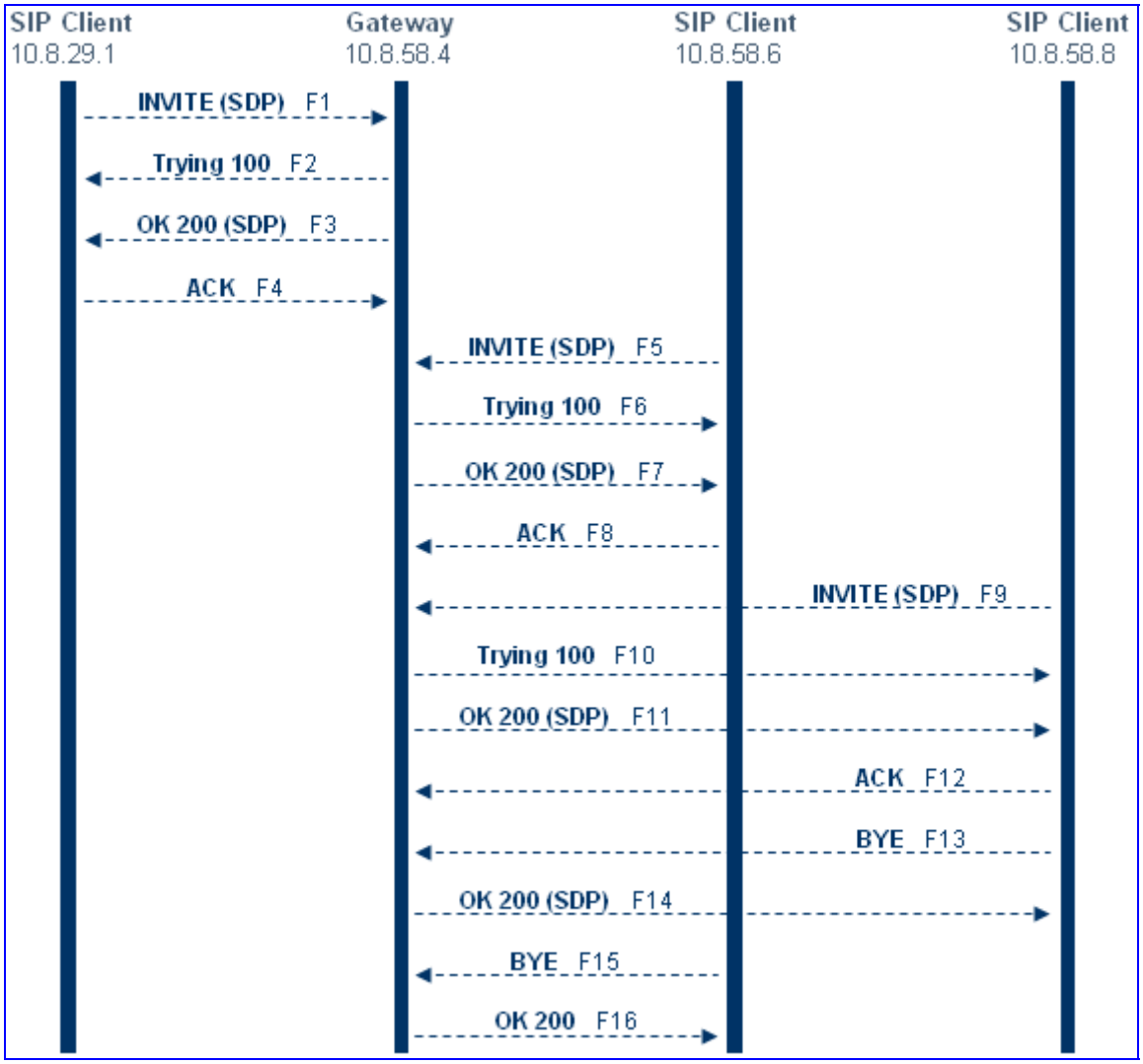
Figure 10-5: Terminating a Conference -- SIP Call Flow



10.1.3 Conference Call Flow Example

The call flow, shown in the following figure, describes SIP messages exchanged between the device (10.8.58.4) and three conference participants (10.8.29.1, 10.8.58.6 and 10.8.58.8).

Figure 10-6: Conference Call Flow Example



1. **SIP MESSAGE 1:** 10.8.29.1:5060 -> 10.8.58.4:5060

```
INVITE sip:conf100@10.8.58.4;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.29.1;branch=z9hG4bKacRHmJhMj
Max-Forwards: 70
From: <sip:100@10.8.8.10>;tag=1c352329022
To: <sip:conf100@10.8.58.4;user=phone>
Call-ID: 1792526528qlax@10.8.29.1
CSeq: 1 INVITE
Contact: <sip:100@10.8.29.1>
Supported: em,100rel,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006
Content-Type: application/sdp
Content-Length: 216
v=0
o=AudiocodesGW 663410 588654 IN IP4 10.8.29.1
s=Phone-Call
c=IN IP4 10.8.29.1
t=0 0
m=audio 6000 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
a=sendrecv
```

2. **SIP MESSAGE 2:** 10.8.58.4:5060() -> 10.8.29.1:5060()

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.8.29.1;branch=z9hG4bKacRHmJhMj
From: <sip:100@10.8.8.10>;tag=1c352329022
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c222574568
Call-ID: 1792526528qlax@10.8.29.1
CSeq: 1 INVITE
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006
Content-Length: 0
```

3. **SIP MESSAGE 3:** 10.8.58.4:5060 -> 10.8.29.1:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.29.1;branch=z9hG4bKacRHmJhMj
From: <sip:100@10.8.8.10>;tag=1c352329022
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c222574568
Call-ID: 1792526528qlax@10.8.29.1
CSeq: 1 INVITE
Contact: <sip:10.8.58.4>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006
Content-Type: application/sdp
Content-Length: 216
v=0
o=AudiocodesGW 820775 130089 IN IP4 10.8.58.4
s=Phone-Call
c=IN IP4 10.8.58.4
t=0 0
m=audio 7160 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
a=sendrecv
```

4. SIP MESSAGE 4: 10.8.29.1:5060 -> 10.8.58.4:5060

```
ACK sip:10.8.58.4 SIP/2.0
Via: SIP/2.0/UDP 10.8.29.1;branch=z9hG4bKacbUrWtRo
Max-Forwards: 70
From: <sip:100@10.8.8.10>;tag=1c352329022
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c222574568
Call-ID: 1792526528qlax@10.8.29.1
CSeq: 1 ACK
Contact: <sip:100@10.8.29.1>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006
Content-Length: 0
```

5. SIP MESSAGE 5: 10.8.58.6:5060 -> 10.8.58.4:5060

```
INVITE sip:conf100@10.8.58.4;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacfowEuut
Max-Forwards: 70
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 1 INVITE
Contact: <sip:600@10.8.58.6>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006
Content-Type: application/sdp
Content-Length: 313
v=0
o=AudiocodesGW 702680 202680 IN IP4 10.8.58.6
s=Phone-Call
c=IN IP4 10.8.58.6
t=0 0
m=audio 6000 RTP/AVP 4 8 0 110 96
a=rtpmap:4 g723/8000
a=fmtp:4 annexa=no
a=rtpmap:8 pcma/8000
a=rtpmap:0 pcmu/8000
a=rtpmap:110 AMR/8000/1
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:30
a=sendrecv
```

6. SIP MESSAGE 6: 10.8.58.4:5060 -> 10.8.58.6:5060

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacfowEuut
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c1673415884
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 1 INVITE
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006
Content-Length: 0
```

7. SIP MESSAGE 7: 10.8.58.4:5060 -> 10.8.58.6:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacfowEuut
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c1673415884
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 1 INVITE Contact: <sip:conf100@10.8.58.4>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006
Content-Type: application/sdp
Content-Length: 236
v=0 o=AudiocodesGW 886442 597756 IN IP4 10.8.58.4
s=Phone-Call
c=IN IP4 10.8.58.4
t=0 0
m=audio 7150 RTP/AVP 4 96
a=rtpmap:4 g723/8000
a=fmtp:4 annexa=no
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:30
a=sendrecv
```

8. SIP MESSAGE 8: 10.8.58.6:5060 -> 10.8.58.4:5060

```
ACK sip:conf100@10.8.58.4 SIP/2.0
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacRRRZPXN
Max-Forwards: 70
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c1673415884
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 1 ACK
Contact: <sip:600@10.8.58.6>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006
Content-Length: 0
```

9. SIP MESSAGE 9: 10.8.58.8:5060 -> 10.8.58.4:5060

```
INVITE sip:conf100@10.8.58.4;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKaczJpxnnv
Max-Forwards: 70
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 1 INVITE
Contact: <sip:800@10.8.58.8>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006
Content-Type: application/sdp Content-Length: 236
v=0
o=AudiocodesGW 558246 666026 IN IP4 10.8.58.8
s=Phone-Call
c=IN IP4 10.8.58.8
t=0 0 m=audio 6000 RTP/AVP 4 96
a=rtpmap:4 g723/8000
a=fmtp:4 annexa=no
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:30
a=sendrecv
```

10. SIP MESSAGE 10: 10.8.58.4:5060 -> 10.8.58.8:5060

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKaczJpxnnv
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c3203015250
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 1 INVITE
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006
Content-Length: 0
```

11. SIP MESSAGE 11: 10.8.58.4:5060 -> 10.8.58.8:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKaczJpxnnv
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c3203015250
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 1 INVITE
Contact: <sip:conf100@10.8.58.4>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006
Content-Type: application/sdp
Content-Length: 236
v=0
o=AudiocodesGW 385533 708665 IN IP4 10.8.58.4
s=Phone-Call
c=IN IP4 10.8.58.4
t=0 0
m=audio 7140 RTP/AVP 4 96
a=rtpmap:4 g723/8000
a=fmtp:4 annexa=no
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:30
a=sendrecv
```

12. SIP MESSAGE 12: 10.8.58.8:5060 -> 10.8.58.4:5060

```
ACK sip:conf100@10.8.58.4 SIP/2.0
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKacisqqyow
Max-Forwards: 70
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c3203015250
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 1 ACK
Contact: <sip:800@10.8.58.8>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006
Content-Length: 0
```

13. SIP MESSAGE 13: 10.8.58.8:5060 -> 10.8.58.4:5060

```
BYE sip:conf100@10.8.58.4 SIP/2.0
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKackSIyGww
Max-Forwards: 70
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c3203015250
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 2 BYE
Contact: <sip:800@10.8.58.8>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006
Content-Length: 0
```

14. SIP MESSAGE 14: 10.8.58.4:5060 -> 10.8.58.8:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKackSIyGww
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c3203015250
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 2 BYE
Contact: <sip:conf100@10.8.58.4>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006
Content-Length: 0
```

15. SIP MESSAGE 15: 10.8.58.6:5060 -> 10.8.58.4:5060

```
BYE sip:conf100@10.8.58.4 SIP/2.0
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacQypxnvl
Max-Forwards: 70
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c1673415884
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 2 BYE
Contact: <sip:600@10.8.58.6>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006
Content-Length: 0
```

16. SIP MESSAGE 16: 10.8.58.4:5060 -> 10.8.58.6:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacQypxnvl
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c1673415884
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 2 BYE
Contact: <sip:conf100@10.8.58.4>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006
Content-Length: 0
```

10.2 Announcement Server

The device supports playing and recording of announcements (local Voice Prompts or HTTP streaming) and playing of Call Progress Tones over the IP network. Three different methods are available for playing and recording announcements:

- NetAnn for playing a single announcement (refer to "NetAnn Interface" on page 483)
- MSCML for playing single or multiple announcements and collecting digits (refer to "MSCML Interface" on page 484)

10.2.1 NetAnn Interface

The device supports playing announcements using NetAnn format (according to RFC 4240).

10.2.1.1 Playing a Local Voice Prompt

To play a single local Voice Prompt, the Application Server (or any SIP user agent) sends a regular SIP INVITE message with SIP URI that includes the NetAnn Announcement Identifier name. For example:

```
INVITE sip:annc@audiocodes.com; play=file://12 SIP/2.0
```

The left part of the SIP URI includes the string 'annc'. In the example above, the device starts playing announcement number 12 from the internal Voice Prompts file (file:// and http://localhost formats are supported). The NetAnn Announcement Identifier string is configured using the *ini* file (parameter NetAnnAnncID) or Web interface (refer to "Configuring the IPmedia Parameters" on page 242). Sending a BYE request terminates the SIP session and stops the playing of the announcement. If the played Voice Prompt reaches its end, the device initiates a BYE message to notify the Application Server that the session has ended.

10.2.1.2 Playing using HTTP/NFS Streaming

To play a single announcement via HTTP or NFS streaming, the Application Server (or any SIP user agent) sends a regular SIP INVITE message with SIP URI that includes the NetAnn Announcement Identifier name. For example:

```
INVITE sip:annc@ac.com;
play=http://server.net/gem/Hello.wav   SIP/2.0
```

The left part of the SIP URI includes the string 'annc' terminated by the IP address of the HTTP server, and the name and path of the file to be played. In the example above, the device starts playing the 'Hello.wav' file that resides in the folder 'server.net/gem'. The NetAnn Announcement Identifier string is configured using the *ini* file (parameter NetAnnAnncID) or Web interface (refer to "Configuring the IPmedia Parameters" on page 242). Sending a BYE request terminates the SIP session and stops the playing of the announcement. If the played announcement reaches its end, the device initiates a BYE message to notify the Application Server that the session is ended.

Notes:

- A 200 OK message is sent only after the HTTP connection is successfully established and the requested file is found. If the file isn't found, a 404 Not Found response is sent.
- To use NFS, the requested file system should be first mounted by using the NFS Servers table, see "Configuring the NFS Settings" on page 74.

10.2.1.3 Supported Attributes

When playing announcements, the following attributes are available:

- **Repeat:** defines the number of times the announcement is repeated. The default value is 1. The valid range is 1 to 1000, or -1 (i.e., repeats the message forever).
- **Delay:** defines the delay (in msec) between announcement repetitions. The default value is 0. The valid range is 1 to 3,600,000.
- **Duration:** defines the total duration (in msec) the announcement(s) are played. The default value is 0 (i.e., no limitation). The valid range is 1 to 3,600,000.

For example:

```
INVITE sip:annc@ac.com;
play=http://server.net/gem/Hello.wav; repeat=5;delay=10000 SIP/2.0
```

10.2.2 MSCML Interface

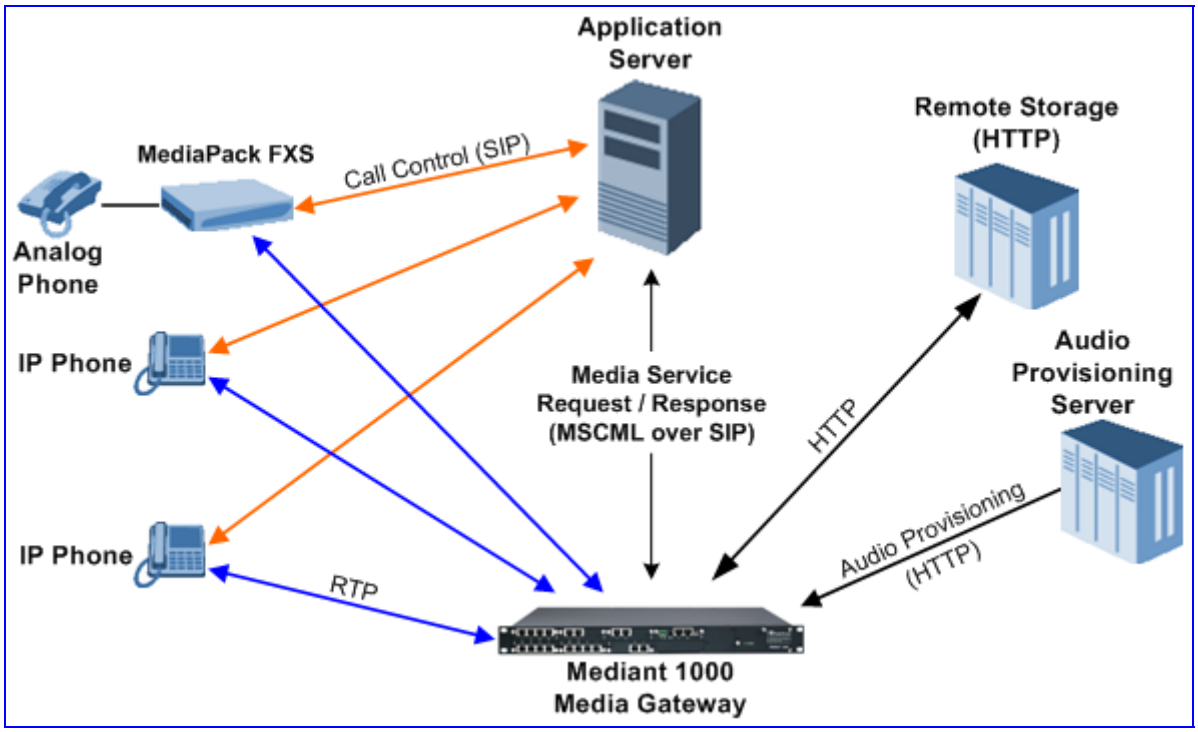
Media Server Control Markup Language (MSCML), according to IETF draft <draft-vandyke-mscml-06.txt>) is a protocol used in conjunction with SIP to provide advanced announcements handling. MSCML is implemented by adding an XML body to existing SIP INFO messages. Only a single message body (containing a single request or response) is allowed per message.

In the current version, the device supports all the Interactive Voice Response (IVR) requirements for playing announcements, collecting digits, and recording (Play, PlayCollect, and PlayRecord).

Note: MSCML is only supported on devices operating with 128-MByte RAM size.

The following figure illustrates standard MSCML application architecture:

Figure 10-7: MSCML Architecture



The architecture comprises the following components:

- **device:** Operating independently, the device controls and allocates its processing resources to match each application's requirements. Its primary role is to handle requests from the Application server for playing announcements and collecting digits.
- **Application Server:** An application platform that controls the call signaling. It interfaces with the device using MSCML. It instructs the media server to play announcements, collect digits and record voice streams.
- **Audio Provisioning Server (APS):** The APS provides the device with a flexible audio package that enables users to easily import audio files, define audio sequences, and include different languages for variable announcement playing.
- **Remote Storage:** An HTTP server that contains less-frequently used voice prompts for playback and to which voice stream recording is performed.
- **IP Phones / MediaPack:** Client applications.

10.2.2.1 Operation

On startup, the device sends a heartbeat packet (a proprietary UDP Ping packet) to the APS. The IP address of the APS to which the device sends the heartbeat packet is defined by the parameter HeartBeatDestIP. After receiving the heartbeat packet, the APS scans its internal database for the IP address (node) of the device (a provision set that includes all necessary audio data is defined for each node). Once found, the APS sends (over HTTP) the provision set to the device. The provision set includes two files: the audio package as a VP.dat file, and an XML file (segments.xml) that contains indices to the announcements stored on the VP.dat file. The two files are stored on the device RAM and are used for playing announcements.

An alternative method uses the AutoUpdate mechanism as described in Automatic Update Mechanism. Both the vp.dat and segments.xml file that were previously created using the APS should be located on an external storage server (HTTP, FTP). At startup, the device fetches the files from the remote storage. By using the AutoUpdate mechanism, the device periodically checks if new files are posted to the remote server and fetches these files.

The Application server communicates with the device using MSCML Requests (sent by the Application server), as shown in the example below:

```
<?xml version="1.0" encoding="utf-8"?>
  <MediaServerControl version="1.0">
    <request>
      ... request body ...
    </request>
  </MediaServerControl>
```

The device uses MSCML Responses (i.e., sent by the device) to reply to the Application server, as shown in the example below:

```
<?xml version="1.0" encoding="utf-8"?>
  <MediaServerControl version="1.0">
    <response>
      ... response body ...
    </response>
  </MediaServerControl>
```

To start an MSCML IVR call, the Application server (or any SIP user agent) sends a regular SIP INVITE message with a SIP URI that includes the MSCML Identifier name. For example:

```
INVITE sip:ivr@audiocodes.com SIP/2.0
```

The left part of the SIP URI includes the MSCML Identifier string 'ivr', which can be configured using the *ini* file (parameter MSCMLID) or Web interface (refer to "Configuring the IPmedia Parameters" on page 242).

After a call is established, SIP INFO messages are used to carry MSCML requests and responses. An INFO message that carries an MSCML body is identified by its content-type header that is set to 'application/mediaservercontrol+xml'.

Note that IVR requests are not queued. Therefore, if a request is received while another is in progress, the device stops the first operation and executes the new request. The device generates a response message for the first request and returns any data collected up to that point. If an application is required to stop a request in progress, it issues a <Stop> request. This request also causes the device to generate a response message.

The device supports basic IVR functions of playing announcements, collecting DTMF digits, and voice stream recording. These services are implemented using the following Request and Response messages:

- <Play> for playing announcements
- <PlayCollect> for playing announcements and collecting digits
- <PlayRecord> for playing announcements and recording voice
- <Stop> for stopping the playing of an announcement

The device sends a Response to each Request that is issued by the Application server.

The <Play>, <PlayCollect>, and <PlayRecord> messages are composed of two sections: Attributes and a Prompt block (the request can contain several different Prompt blocks). The Attributes section includes several request-specific parameters. The Prompt block section itself is also composed of two sections: prompt-specific parameters and audio segments (audio / variable). The (optional) prompt-specific parameters include:

- *locale*: defines the language in which the prompt block is played (supported for local files only). For detailed information on language usage, refer to the *Audio Provisioning Server User's Manual* (LTRT-971xx).
- *baseurl*: defines a URL address that functions as a prefix to all audio segment URLs in the Prompt block.

The Prompt block contains references to one or more audio segments. The following audio segment types are available:

- **Physical Audio Segments:** These are physical audio files that are located either locally (on-blade) or on an external HTTP server. If the file is located on-blade, the reference to it is by using one of the following syntaxes:

  'file://x', 'file:///x', 'file:////x' or 'http://localhost/x'

  Where *x* stands for the file identifier (the ID or alias given by the APS server for local files; or the file's URL in for HTTP streaming).

- **Variables:** These are audio segments whose value is determined at run time. They are defined in the request as a <type, subtype, value> tuple. The device transforms the variable data to voice. To support variable playing, APS server support is mandatory. Available variable types are (subtypes in parenthesis): date, duration, month, money (USD), number (crd, ord), digit (gen, ndn) silence, string, time (t12, t24) and weekday.
  It is also possible to store audio files that are required to play supported types of phrases (e.g., dates and times) on an off-board system. This is beneficial in scenarios where the device's on-board storage limit has been reached, and thus, additional languages and audio can be stored off-board.

- **Sequences:** These are audio segments that consist of physical audio files and variables. These sequences can be defined using the APS server.

10.2.2.2 Playing Announcements

A <Play> request is used to play an announcement to the caller. Each <Play> request contains a single Prompt block and the following request-specific parameters:

- *id*: an optional random number used to synchronize request and response.
- *prompturl*: a specific audio file URL that is used in addition to the references in the Prompt block. This audio file is the first to be played.

An example of an MSCML <Play> Request that includes local and streaming audio files as well as variables is shown below:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
    <request>
          <play id=”123”>
                <prompt>
                      <audio url="http://localhost/1"/>
                      <variable type="digits" value="284"/>
                      <variable type="silence" value="1"/>
                      <audio url="http://10.3.0.2/aa.wav"/>
                <audiourl="nfs://10.3.0.3/prov data/bb.wav"/>
                </prompt>
          </play>
    </request>
</MediaServerControl>
```

10.2.2.3 Playing Announcements and Collecting Digits

The <PlayCollect> request is used to play an announcement to the caller and to then collect entered DTMF digits. The play part of the <PlayCollect> request is identical to the <Play> request. The collect part includes an expected digit map. The collected digits are continuously compared to the digit map. Once a match is found, the collected digits are sent in a <PlayCollect> response. The digit map should be in MGCP format (the type value must be set to ‘mgcpdigitmap’).

For example:

```
<regex type="mgcpdigitmap" value="([0-1]xxx)">
</regex>
```

Each <PlayCollect> request contains the following request-specific parameters in addition to the Prompt block (all parameters are optional):

- *id*: an optional random number used to synchronize request and response.
- *prompturl:* a specific audio file URL that is used in addition to the references in the prompt block. This audio file is the first to be played.
- *barge*: if set to ‘NO’, DTMF digits received during announcement playback are ignored. If set to ‘YES’, DTMF digits received during announcement playback stop the playback and start the digit collection phase.
- *firstdigittimer:* defines the amount of time (in milliseconds) the user does not enter any digits, after which a response is sent indicating timeout.
- *interdigittimer:* defines the amount of time (in milliseconds) the user does not enter any digits after the first DTMF digit is received, after which a response is sent indicating timeout.

- *extradigittimer:* used to enable the following:
  - Detection of command keys (ReturnKey and EscapeKey).
  - Not report the shortest match. MGCP Digitmap searches for the shortest possible match. This means that if a digitmap of (123 | 1234) is defined, once the user enters 123, a match is found and a response is sent. If ExtraDigitTimer is defined, the match can also be 1234 because the device waits for the next digits. To use ExtraDigitTimer, it must be defined in the request and you must add a "T" to the Digitmap (for example, 'xxT'). The ExtraDigitTimer is only used when a match is found. Before a match is found, the timer used is the InterDigitTimer. Therefore, if the ExtraDigitTimer expires, a "match" response reason is reported -- never a "timeout".
- *maxdigits:* defines the maximum number of collected DTMF digits after which the device sends a response.
- *cleardigits:* defines whether or not the device clears the digit buffer between subsequent requests.
- *returnkey*: defines a specific digit (including '*' and '#') which (when detected during a collection) stops the collection and initiates a response (that includes all digits collected up to that point) to be sent.
- *escapekey:* defines a specific digit (including '*' and '#') which (when detected during a collection) stops the collection and initiates a response (with no collected digits) to be sent.

An example is shown below of an MSCML <PlayCollect> Request that includes a sequence with variables and an MGCP digit map:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
          <request>
                <playcollect id="6379" barge="NO" returnkey="#">
                      <prompt>
                            <audio url="http://localhost/1">
                            <variable type="silence" value="1"/>
                            <variable type="date" subtype="mdy"
value="20041210"/>
                      </prompt>
                      <regex type="mgcpdigitmap" value="([0-
1]xxx)">
                      </regex>
                </playcollect>
          </request>
</MediaServerControl>
```

An example is shown below of an MSCML <PlayCollect> Response:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
          <response request="playcollect" id="6478" code="200"
text="OK" digits="4563">
          </response>
</MediaServerControl>
```

10.2.2.4 Playing Announcements and Recording Voice

The <PlayRecord> request is used to play an announcement to the caller and to then record the voice stream associated with that caller. The play part of the <PlayRecord> request is identical to the <Play> request. The record part includes a URL to which the voice stream is recorded. This URL refers to an HTTP server.

Each <PlayRecord> request contains the following request-specific parameters in addition to the Prompt block (all parameters except 'recurl' are optional):

- *id*: an optional random number used to synchronize request and response.
- *prompturl:* a specific audio file URL that is used in addition to the references in the prompt block. This audio file is the first to be played.
- *barge*: if set to 'NO', DTMF digits received during announcement playback are ignored. If set to 'YES', DTMF digits received during announcement playback stop the playback and start the recording phase.
- *cleardigits:* defines whether or not the device clears the digit buffer between subsequent requests.
- *escapekey:* defines a specific digit (including '*' and '#') which (when detected during any phase) stops the request and initiates a response.
- *recurl*: the URL on the external storage server to which the RTP stream is sent for recording. This is a mandatory parameter.
- *mode:* defines if the recording 'overwrites' the existing file or 'appends' to it.
- *initsilence:* defines how long to wait for initial speech input before terminating the recording. This parameter may take an integer value in milliseconds.
- *endsilence:* defines how long the device waits after speech has ended to stop the recording. This parameter may take an integer value in milliseconds.
- *duration*: the total time in milliseconds for the entire recording. Once this time expires, recording stops and a response is generated.
- *recstopmask:* defines a digit pattern to which the device compares digits detected during the recording phase. If a match is found, recording stops and a response is sent.

An example is shown below of an MSCML <PlayRecord> Request:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
          <request>
                <playrecord id="75899" barge="NO"
Recurl=nfs://10.11.12.13/save/recordings/11.wav>
                      <prompt>
                            <audio url="nfs://100.101.102.103/45">
                            <variable type="date" subtype="mdy"
value="20041210"/>
                      </prompt>
                </playrecord>
          </request>
</MediaServerControl>
```

An example is shown below of an MSCML <PlayRecord> Response:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
          <response request="playrecord" id="75899" code="200"
text="OK" reclength="15005">
          </response>
</MediaServerControl>
```

10.2.2.5 Stopping the Playing of an Announcement

The Application server issues a <stop> request when it requires that the device stops a request in progress and not initiate another operation. The only (optional) request-specific parameter is id.

The device refers to a SIP re-INVITE message with hold media (c=0.0.0.0) as an implicit <Stop> request. The device immediately terminates the request in progress and sends a response.

An example is shown below of an MSCML <Stop> Request:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
          <request>
                <stop id="123">
                </stop>
          </request>
</MediaServerControl>
```

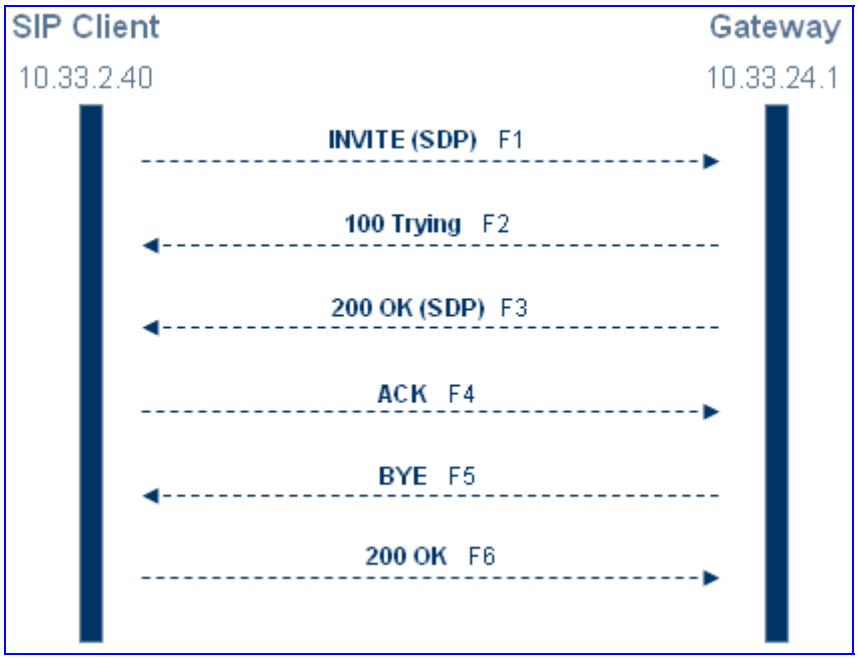
10.2.2.6 Relevant Parameters

The following parameters (described in "Media Server Parameters" on page 342) are used to configure the MSCML:

- AmsProfile = 1 (mandatory)
- AASPackagesProfile = 3 (mandatory)
- VoiceStreamUploadMethod = 1 (mandatory)
- EnableVoiceStreaming = 1 (mandatory)
- MSCMLID (default="ivr")
- AmsPrimaryLanguage (default="eng")
- AmsSecondaryLanguage (default="heb")
- When using APS:
  - HeartBeatDestIP (refer to "System Parameters" on page 305)
  - HeartBeatDestPort
  - HeartBeatIntervalmsec
- When using AutoUpdate:
  - VPFileURL
  - APSSegmentsFileUrl
  - AutoUpdateFrequency / AutoUpdatePredefinedTime

10.2.3 Announcement Call Flow Example

The call flow, shown in the following figure, describes SIP messages exchanged between the device (10.33.24.1) and a SIP client (10.33.2.40) requesting to play local announcement #1 (10.8.25.17) using AudioCodes proprietary method.



1. **SIP MESSAGE 1:** 10.33.2.40:5060 -> 10.33.24.1:5060

```
INVITE sip:annc@10.33.24.1;play=http://10.3.0.2/hello.wav;repeat=2
SIP/2.0
Via: SIP/2.0/UDP 10.33.2.40;branch=z9hG4bKactXhKPQT
Max-Forwards: 70
From: <sip:103@10.33.2.40>;tag=1c2917829348
To: <sip:annc@10.33.24.1>
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 INVITE
Contact: <sip:103@10.33.2.40>
Supported: em,100rel,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-4.0 GA/v.4.0 GA
Content-Type: application/sdp
Content-Length: 215
v=0
o=AudiocodesGW 377662 728960 IN IP4 10.33.41.52
s=Phone-Call
c=IN IP4 10.33.41.52
t=0 0
m=audio 4030 RTP/AVP 4 0 8
a=rtpmap:4 g723/8000
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=ptime:30
a=sendrecv
```

2. SIP MESSAGE 2: 10.33.24.1:5060 -> 10.33.2.40:5060

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.33.2.40;branch=z9hG4bKactXhKPQT
From: <sip:103@10.33.2.40>;tag=1c2917829348
To: <sip:annc@10.33.24.1>;tag=1c1528117157
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 INVITE
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-TrunkPack 1610/v.4.60AOH.006.002D
Content-Length: 0
```

3. SIP MESSAGE 3: 10.33.24.1:5060 -> 10.33.2.40:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.33.2.40;branch=z9hG4bKactXhKPQT
From: <sip:103@10.33.2.40>;tag=1c2917829348
To: <sip:annc@10.33.24.1>;tag=1c1528117157
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 INVITE Contact: <sip:10.33.24.1>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-TrunkPack 1610/v.4.60AOH.006.002D
Content-Type: application/sdp
Content-Length: 165
v=0
o=AudiocodesGW 355320 153319 IN IP4 10.33.24.1
s=Phone-Call
c=IN IP4 10.33.24.1
t=0 0
m=audio 7170 RTP/AVP 0
a=rtpmap:0 pcmu/8000
a=ptime:20
a=sendrecv
```

4. SIP MESSAGE 4: 10.33.2.40:5060 -> 10.33.24.1:5060

```
ACK sip:10.33.24.1 SIP/2.0
Via: SIP/2.0/UDP 10.33.2.40;branch=z9hG4bKacnNUEeKP
Max-Forwards: 70
From: <sip:103@10.33.2.40>;tag=1c2917829348
To: <sip:annc@10.33.24.1>;tag=1c1528117157
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 ACK
Contact: <sip:103@10.33.2.40>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-4.0 GA/v.4.0 GA
Content-Length: 0
```

5. **SIP MESSAGE 5:** 10.33.24.1:5060 -> 10.33.2.40:5060

```
BYE sip:103@10.33.2.40 SIP/2.0
Via: SIP/2.0/UDP 10.33.24.1;branch=z9hG4bKacFhtFbFR
Max-Forwards: 70
From: <sip:annc@10.33.24.1>;tag=1c1528117157
To: <sip:103@10.33.2.40>;tag=1c2917829348
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 BYE
Contact: <sip:10.33.24.1>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-TrunkPack
1610/v.4.60AOH.006.002D
Content-Length: 0
```

6. **SIP MESSAGE 6:** 10.33.2.40:5060 -> 10.33.24.1:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.33.24.1;branch=z9hG4bKacFhtFbFR
From: <sip:annc@10.33.24.1>;tag=1c1528117157
To: <sip:103@10.33.2.40>;tag=1c2917829348
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 BYE
Contact: <sip:103@10.33.2.40>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-4.0 GA/v.4.0 GA
Content-Length: 0
```

10.3 IP-to-IP Transcoding

Transcoding is a technology that is used to bridge (translate) between two remote *network* locations, each of which uses a different coder and/or a different DTMF and fax transport types. The device supports IP-to-IP Transcoding. It creates a Transcoding call that is similar to a dial-in, two-party conference call. The SIP URI in the INVITE message is used as a Transcoding service identifier. The Transcoding identifier can be configured using the *ini* file (parameter TranscodingID) or Web interface (refer to "Configuring the IPmedia Parameters" on page 242.

It is assumed that the device is controlled by a third-party, Application server (or any SIP user agent) that instructs the device to start an IP Transcoding call by sending two SIP INVITE messages with SIP URI that includes the Transcoding Identifier name. For example:

```
Invite sip:trans123@audiocodes.com SIP/2.0
```

The left part of the SIP URI includes the Transcoding ID (the default string is 'trans') and is terminated by a unique number (123). The device immediately sends a 200 OK message in response to each INVITE.

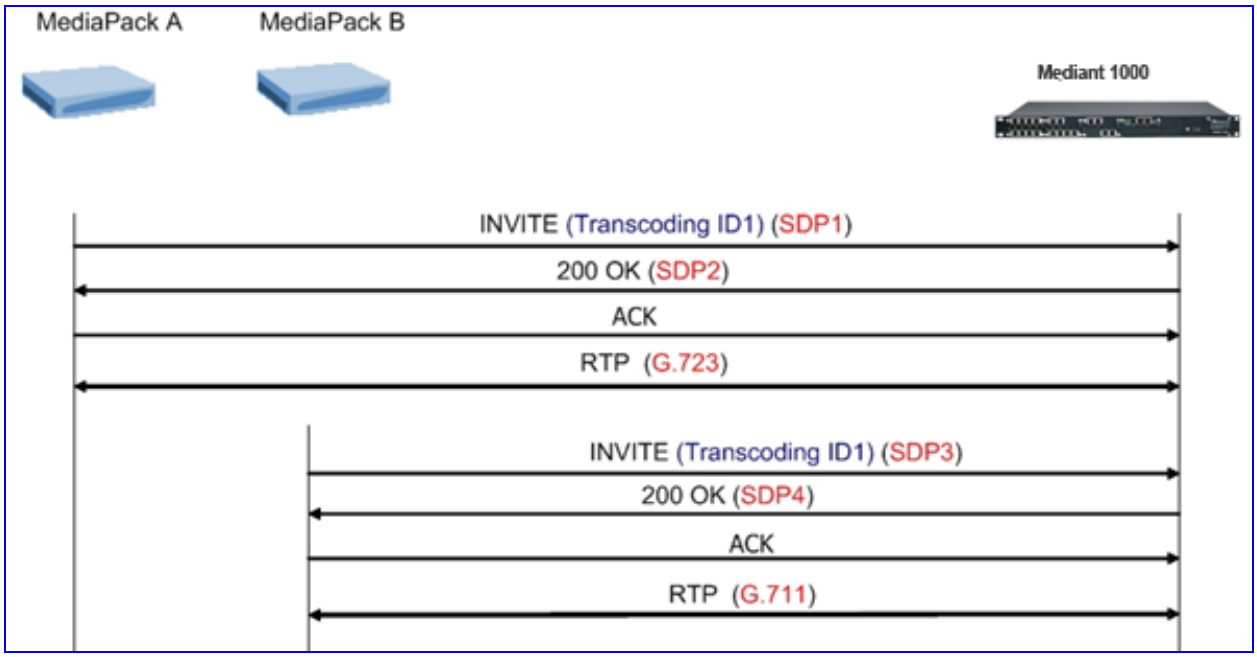
Each of the Transcoding SIP call participants can use a different VoIP coder and a different DTMF transport type, negotiated with the device using common SIP negotiation.

Sending a BYE request to the device by any of the participants, terminates the SIP session and removes it from the Transcoding session. The second BYE from the second participant ends the Transcoding session and releases its resources.

The device uses two media (DSP) channels for each call, thereby reducing the number of available Transcoding sessions to half of the defined value for MediaChannels. To limit the number of resources available for Transcoding, use the *ini* file parameter MediaChannels or Web interface (refer to "Configuring the IPmedia Parameters" on page 242). For example, if MediaChannels is 40, only 20 Transcoding sessions are available.

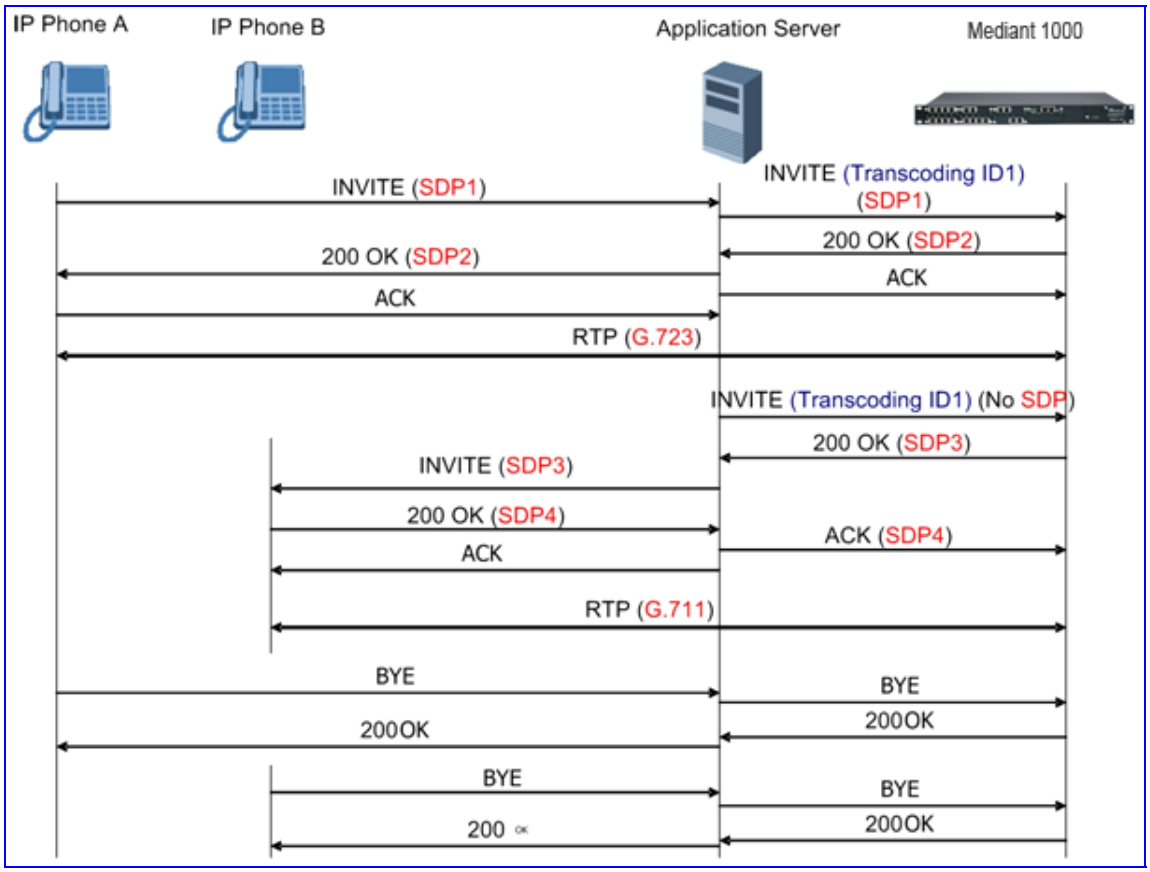
The figure below illustrates an example of a direct connection to a device:

Figure 10-8: Direct Connection (Example)



The figure below illustrates an example of implementing an Application server:

Figure 10-9: Using an Application Server



11 Tunneling Applications

This section discusses TDM and QISG tunneling, supported by the device.

11.1 TDM Tunneling

The device's TDM Tunneling feature allows you to tunnel groups of digital trunk spans or timeslots (B-channels) over the IP network. TDM Tunneling utilizes the device's internal routing (without Proxy control) capabilities to receive voice and data streams from TDM (E1/T1/J1/) spans or individual timeslots, convert them into packets, and then transmit them over the IP network (using point-to-point or point-to-multipoint device distributions). A device opposite it (or several devices when point-to-multipoint distribution is used) converts the IP packets back into TDM traffic. Each timeslot can be targeted to any other timeslot within a trunk in the opposite device.

When TDM Tunneling is enabled ('Enable TDM Tunneling' parameter is set to 'Enable' on the originating device -- refer to "Configuring the Digital Gateway Parameters" on page 235), the originating device automatically initiates SIP calls from all enabled B-channels belonging to the E1/T1/J1 spans that are configured with the protocol type 'Transparent' (for ISDN trunks) or 'Raw CAS' (for CAS trunks). The called number of each call is the internal phone number of the B-channel from where the call originates. The 'IP to Trunk Group Routing' table (refer to "IP to Trunk Group Routing" on page 204) is used to define the destination IP address of the terminating device. The terminating device automatically answers these calls if its E1/T1 protocol type is set to 'Transparent' (ProtocolType = 5) or 'Raw CAS' (ProtocolType = 3 for T1 and 9 for E1) and the parameter ChannelSelectMode is set to 0 (By Phone Number).

Note: It's possible to configure both devices to also operate in symmetric mode. To do so, set EnableTDMOverIP to 1 and configure the 'Tel to IP Routing' tables in both devices. In this mode, each device (after it's reset) initiates calls to the second device. The first call for each B-channel is answered by the second device.

The device continuously monitors the established connections. If for some reason, one or more calls are released, the device automatically re-establishes these 'broken' connections. In addition, when a failure in a physical trunk or in the IP network occurs, the device re-establishes the tunneling connections when the network is restored.

Note: It's recommended to use the keep-alive mechanism for each connection, by activating the 'session expires' timeout and using Re-INVITE messages.

By utilizing the 'Profiles' mechanism (refer to "Configuring the Profile Definitions" on page 212), you can configure the TDM Tunneling feature to choose different settings based on a timeslot or groups of timeslots. For example, you can use low-bit-rate vocoders to transport voice and 'Transparent' coder to transport data (e.g., for D-channel). You can also use Profiles to assign ToS (for DiffServ) per source -- a timeslot carrying data or signaling is assigned a higher priority value than a timeslot carrying voice.

For tunneling of E1/T1 CAS trunks, set the protocol type to 'Raw CAS' (ProtocolType = 3 / 9) and enable RFC 2833 CAS relay mode ('CAS Transport Type' parameter is set to 'CAS RFC2833 Relay' -- refer to "Configuring the Voice Settings" on page 81).

Note: For TDM over IP, the 'Caller ID Transport Type' parameter must be set to 'Disable', i.e., transparent (refer to "Configuring the Fax / Modem / CID Settings" on page 84).

Below is an example of *ini* files for two devices implementing TDM Tunneling for four E1 spans. Note that in this example both devices are dedicated to TDM tunneling.

Terminating Side:

```
EnableTDMOverIP = 1
;E1 TRANSPARENT 31
ProtocolType 0 = 5
ProtocolType 1 = 5
ProtocolType 2 = 5
ProtocolType 3 = 5
[PREFIX]
FORMAT PREFIX_Index = PREFIX_DestinationPrefix,
PREFIX DestAddress, PREFIX SourcePrefix, PREFIX ProfileId,
PREFIX MeteringCode, PREFIX DestPort;
Prefix 1 = '*,10.8.24.12';
[\PREFIX]

;IP address of the device in the opposite
;location
;Channel selection by Phone number.
ChannelSelectMode = 0
;Profiles can be used do define different coders per B-channels
;such as Transparent
;coder for B-channels (timeslot 16) that carries PRI ;signaling.
[TrunkGroup]
FORMAT TrunkGroup Index = TrunkGroup TrunkGroupNum,
TrunkGroup_FirstTrunkId, TrunkGroup_LastTrunkId,
TrunkGroup FirstBChannel, TrunkGroup LastBChannel,
TrunkGroup FirstPhoneNumber, TrunkGroup ProfileId,
TrunkGroup Module;
TrunkGroup 1 = 0,0,0,1,31,1000,1;
TrunkGroup 1 = 0,1,1,1,31,2000,1;
TrunkGroup 1 = 0,2,2,1,31,3000,1;
TrunkGroup 1 = 0,3,3,1,31,4000,1;
TrunkGroup 1 = 0,0,0,16,16,7000,2;
TrunkGroup 1 = 0,1,1,16,16,7001,2;
TrunkGroup 1 = 0,2,2,16,16,7002,2;
TrunkGroup 1 = 0,3,3,16,16,7003,2;
[/TrunkGroup]
[CoderName]
FORMAT CoderName Index = CoderName Type, CoderName PacketInterval,
CoderName rate, CoderName PayloadType, CoderName Sce;
CoderName 0 = 'g7231';
CoderName 1 = 'Transparent';
CoderName 5 = 'g7231';
CoderName 6 = 'Transparent';
[/CoderName]
[TelProfile]
FORMAT TelProfile_Index = TelProfile_ProfileName,
TelProfile TelPreference, TelProfile CodersGroupID,
TelProfile IsFaxUsed, TelProfile JitterBufMinDelay,
TelProfile_JitterBufOptFactor, TelProfile_IPDiffServ,
TelProfile SigIPDiffServ, TelProfile DtmfVolume,
TelProfile InputGain, TelProfile VoiceVolume,
TelProfile_EnableReversePolarity,
```

```
TelProfile EnableCurrentDisconnect,
TelProfile EnableDigitDelivery, TelProfile EnableEC,
TelProfile_MWIAnalog, TelProfile_MWIDisplay,
TelProfile FlashHookPeriod, TelProfile EnableEarlyMedia,
TelProfile ProgressIndicator2IP;
TelProfile 1 = voice,$$,1,$$,$$,$$,$$,$$,$$,$$;
TelProfile 2 = data,$$,2,$$,$$,$$,$$,$$,$$,$$;
[\TelProfile]
```

Originating Side:

```
;E1_TRANSPARENT_31
ProtocolType 0 = 5
ProtocolType 1 = 5
ProtocolType_2 = 5
ProtocolType 3 = 5
;Channel selection by Phone number.
ChannelSelectMode = 0
[TrunkGroup]
FORMAT TrunkGroup Index = TrunkGroup TrunkGroupNum,
TrunkGroup FirstTrunkId, TrunkGroup LastTrunkId,
TrunkGroup FirstBChannel, TrunkGroup LastBChannel,
TrunkGroup FirstPhoneNumber, TrunkGroup ProfileId,
TrunkGroup Module;
TrunkGroup 0 = 0,0,0,1,31,1000,1;
TrunkGroup 0 = 0,1,1,1,31,2000,1;
TrunkGroup 0 = 0,2,2,1,31,3000,1;
TrunkGroup 0 = 0,3,1,31,4000,1;
TrunkGroup 0 = 0,0,0,16,16,7000,2;
TrunkGroup 0 = 0,1,1,16,16,7001,2;
TrunkGroup 0 = 0,2,2,16,16,7002,2;
TrunkGroup 0 = 0,3,3,16,16,7003,2;
[\TrunkGroup]
[CoderName]
FORMAT CoderName Index = CoderName Type, CoderName PacketInterval,
CoderName rate, CoderName PayloadType, CoderName Sce;
CoderName 1 = 'g7231';
CoderName 2 = 'Transparent';
[\CoderName]
[TelProfile]
FORMAT TelProfile Index = TelProfile ProfileName,
TelProfile TelPreference, TelProfile CodersGroupID,
TelProfile_IsFaxUsed, TelProfile_JitterBufMinDelay,
TelProfile JitterBufOptFactor, TelProfile IPDiffServ,
TelProfile SigIPDiffServ, TelProfile DtmfVolume,
TelProfile_InputGain, TelProfile_VoiceVolume,
TelProfile EnableReversePolarity,
TelProfile EnableCurrentDisconnect,
TelProfile_EnableDigitDelivery, TelProfile_EnableEC,
TelProfile MWIAnalog, TelProfile MWIDisplay,
TelProfile FlashHookPeriod, TelProfile EnableEarlyMedia,
TelProfile_ProgressIndicator2IP;
TelProfile 1 = voice,$$,1,$$,$$,$$,$$,$$,$$,$$
TelProfile 2 = data,$$,2,$$,$$,$$,$$,$$,$$,$$
[\TelProfile]
```

11.2 QSIG Tunneling

The device supports QSIG tunneling over SIP according to IETF draft 'Tunnelling of QSIG over SIP' (draft-elwell-sipping-qsig-tunnel-03) and the ECMA-355/ISO/IEC 22535 standard. This method enables all QSIG messages to be sent as raw data in corresponding SIP messages using a dedicated message body. This mechanism is useful for two QSIG subscribers (connected to the same or different QSIG PBX) to communicate with each other over an IP network. Tunneling is supported in both directions (Tel-to-IP and IP-to-Tel).

The term tunneling means that messages are transferred 'as is' to the remote side without being converted (QSIG→SIP→QSIG). The advantage of tunneling over QSIG-to-SIP interworking is that by using interworking, QSIG functionality can only be partially achieved. When tunneling is used, all QSIG capabilities are supported, whereas the tunneling medium (the SIP network) does not need to process these messages.

QSIG messages are transferred in SIP messages in a separate Multipurpose Internet Mail Extensions (MIME) body. Therefore, if a message contains more than one body (e.g., SDP and QSIG), multipart MIME must be used. The Content-Type of the QSIG tunneled message is 'application/QSIG'. In addition, the device adds a Content-Disposition header in the following format:

```
Content-Disposition: signal; handling=required.
```

- **Call setup (originating device):** The QSIG SETUP request is encapsulated in the SIP INVITE message without being altered. After the SIP INVITE request is sent, the device doesn't encapsulate the subsequent QSIG message until a SIP 200 OK response is received. If the originating device receives a 4xx, 5xx, or 6xx response, it disconnects the QSIG call with a 'no route to destination' cause.
- **Call setup (terminating device):** After the terminating device receives a SIP INVITE request with a 'Content-Type: application/QSIG', it sends the encapsulated QSIG SETUP message to the Tel side and sends a 200 OK response (no 1xx response is sent) to IP. The 200 OK response includes an encapsulated QSIG CALL PROCEEDING message (without waiting for a CALL PROCEEDING message from the Tel side). If tunneling is disabled and the incoming INVITE includes a QSIG body, a 415 response is sent.
- **Mid-call communication:** After the SIP connection is established, all QSIG messages are encapsulated in SIP INFO messages.
- **Call tear-down:** The SIP connection is terminated once the QSIG call is complete. The RELEASE COMPLETE message is encapsulated in the SIP BYE message that terminates the session.

To enable QSIG tunneling, set the parameter EnableQSIGTunneling to 1 on both the originating and terminating devices, and the parameter ISDNDuplicateQ931BuffMode to 128 (duplicate all messages) (both parameters are described in "ISDN and CAS Interworking-Related Parameters" on page 348).

12 OSN Server Hardware Installation

This section is intended for customers who have purchased the Mediant 1000 media gateway and wish to install the optional Mediant 1000 OSN (Open Solution Network) server functionality.

The Mediant 1000 chassis can house a plug-in OSN Server module for hosting third-party, VoIP applications such as IP-PBX, Pre-Paid, and IP-PBX redundancy. The OSN server is a standalone entity, integrated within the Mediant 1000 gateway, using a separate Ethernet interface and IP configuration from that used by the gateway.

Note: This section is applicable only to Mediant 1000.

12.1 OSN Server Modules

The Mediant 1000 OSN Server package is composed of three modules, which need to be installed in the Mediant 1000 chassis:

- Connection module (CM)

Figure 12-1: Connection Module (CM)

- iPMX module

Figure 12-2: iPMX Module

- Hard Drive module (HDMX)

Figure 12-3: Hard Drive Module (HDMX)

Warning: Before installing the Mediant 1000 OSN Server modules, ensure that the Mediant 1000 is disconnected from the power supply. These modules are not hot-swappable and damage to these modules can occur if replaced under voltage.

12.2 Required Working Tools

The following tools are required for installing the OSN Server module:

- Phillips screwdriver
- Flathead screwdriver
- Wire cutter

12.3 Installing the CM Module

The Connection Module (CM) is installed on the front panel of the Mediant 1000, as described in the following procedure:

➢ To install the CM module, take these 4 steps:

1. On the Mediant 1000 front panel, use a Phillips screwdriver to remove the black metal cover plate from the slot located below the module labeled **CPU**, as shown in the figure below:

Figure 12-4: Mediant 1000 Front Panel

2. Insert the CM module into the empty slot (below the CPU), with the plain side of the Printed Circuit Board (PCB) facing up. Ensure the PCB slides into the slot rails, by aligning the CM with the rails in the slot.

Figure 12-5: Inserting CM Module

3. Gently push the CM module into the slot until it is fully inserted.
4. Using a flathead screwdriver, tighten the module's mounting pins.

12.4 Installing the iPMX Module

The iPMX module is installed on the rear panel of the Mediant 1000, as described in the following procedure:

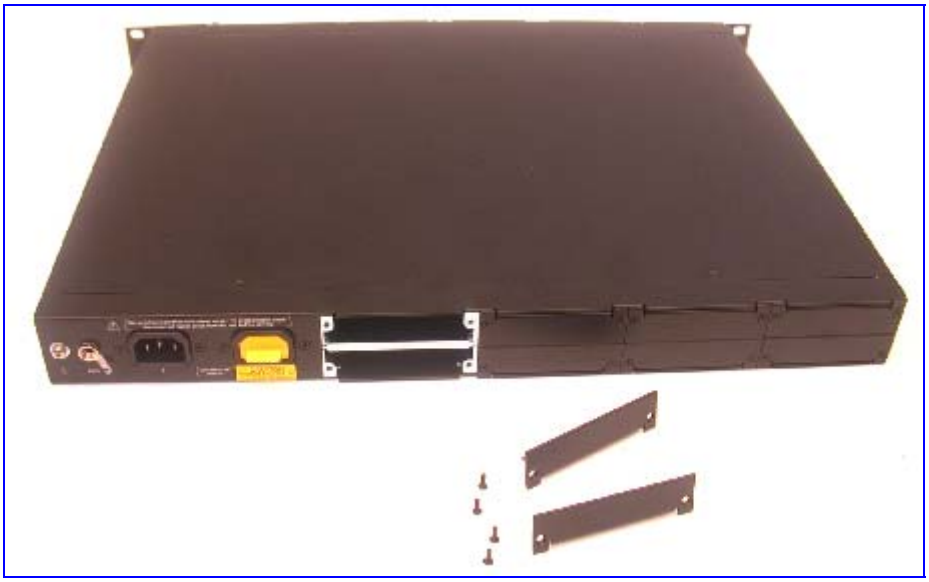
➢ To install the iPMX module, take these 7 steps:

1. Place the Mediant 1000 so that the rear panel is facing you, as shown in the figure below.

Figure 12-6: Mediant 1000 Rear Panel

2. Remove the black metal cover plates in the first and second slots located on the right side of the power connection, as shown in the figure below.

Figure 12-7: Mediant 1000 with Cover Plates Removed



3. Use the cutter tool to remove the small metal strip between the upper and lower slots, as shown in the figure below.

Figure 12-8: Mediant 1000 with Cutter Tool

4. Insert the iPMX module into the first slot, closest to the power connection, as shown in the figure below.

Figure 12-9: Inserting iPMX Module

5. Push the iPMX module into the slot and press on it firmly to ensure it has been fully inserted.
6. Using a flathead screwdriver, tighten the module's two captive mounting screws located on the bottom right and left corners.
7. Using a Philips screwdriver, tighten the module's two Philips screws located on the top right and left corners.

12.5 Installing the HDMX Module

The Hard Drive module (HDMX) is installed on the rear panel of the Mediant 1000, as described in the following procedure:

➢ To install the HDMX module, take these 6 steps:

1. Place the Mediant 1000 so that the rear panel is facing you.
2. Remove the black metal cover plates in the first and second slots located on the right side of the power connection.
3. Use the cutter tool to remove the small metal strip between the upper and lower slots.
4. Insert the HDMX module into the second slot, as shown in the figure below.

Figure 12-10: Inserting HDMX Module

5. Push the HDMX module into the slot and press on it firmly to ensure it has been fully inserted.
6. Using a flathead screwdriver, tighten the module's mounting pins.

12.6 Replacing the iPMX Module's Lithium Battery

The iPMX module is equipped with a 3-volt CR-1225 Lithium battery (AudioCodes product number: ACL P/N RBAT00001). Typically, battery life is estimated at two years. However, for various reasons, the battery may last for a shorter duration.

Warnings:

- When replacing the battery, all BIOS settings revert to factory defaults.
- When removing and inserting the battery, be careful not to touch other components on the iPMX printed circuit board (PCB) with the extracting tool. This may cause irreversible damage to the iPMX module.
- Dispose of used batteries according to the manufacturer's instructions. Failure to do so could result in environmental damage.
- The Lithium battery must only be replaced with an identical or equivalent battery, as recommended by the manufacturer.

Electrical Component Sensitivity

Electronic components on printed circuit boards are extremely sensitive to static electricity. Normal amounts of static electricity generated by clothing can damage electronic equipment. To reduce the risk of damage due to electrostatic discharge (ESD) when installing or servicing electronic equipment, it is recommended that anti-static earthing straps and mats be used.

The following procedure describes how to replace the Lithium battery in the iPMX module.

➢ To replace the Lithium battery in the iPMX, take these 6 steps:

1. Remove the iPMX module from the slot in which it's housed in the Mediant 1000 rear panel, by performing the following:
   a. Using a flathead screwdriver, loosen the module's two lower mounting captive screws.
   b. Using a Philips screwdriver, loosen the two upper screws.
   c. Holding the two mounting captive screws, gently pull the module out of the slot.
2. Flip the module over so that it lies face down with the PCB visible.
3. Locate the Lithium battery in its battery holder on the circuit board.

4. Using a tweezer-like tool (or small flathead screwdriver), carefully leverage the battery out of the battery holder. Be careful not to touch other components on the board with your tool.

Figure 12-11: Removing Lithium Battery from iPMX Module

5. For installing the new battery, simply push the battery into the battery holder using your fingers. Ensure that you install the battery in the correct orientation such that the positive side is facing up (i.e., the side containing the battery description is visible).

6. Re-insert the iPMX module into the slot of the Mediant 1000 chassis as described in the previous section.

13 Installing Windows XP on the OSN Server

The embedded PC on Mediant 1000 (OSN module) can be ordered with Windows XP pre-installed. However, if requested, the OSN server can be supplied without an installed Windows operating system. This section provides step-by-step procedures on installing Microsoft Windows XP on the OSN hard drive. (Although not tested, these procedures should be applicable to Windows 2003.)

The mechanism used to install Windows on the OSN module is based on "unattended" or "headless" mechanisms provided by the Microsoft OS. Several tools, both commercially and freeware, exist that can assist in converting a Microsoft Windows installation CD into an ISO image (*.iso file), which can be used for unattended installation. This guideline is based on the freeware utility "Windows unattended CD creator" available from the following website: http://www.vitaligraf.de/en/?Projects:Windows_Unattended_CD_Creator.

13.1 Requirements

The required hardware for installing Windows XP includes the following:

- Mediant 1000 with installed OSN module
- Mediant 1000 OSN serial cable
- USB hub
- USB CD/DVD drive
- USB keyboard

The required software for installing Windows XP includes the following:

- HyperTerminal or equivalent terminal emulation program
- Notepad or equivalent text editor
- Original Windows installation disk
- Windows Unattended CD Creator (WuCDC)
- Mediant 1000 OSN LAN drivers (Intel 82551IT)
- Mediant 1000 OSN sound card drivers (Sigmatel C-major Audio)

13.2 Windows XP Installation Procedures

13.2.1 Stage 1: Creating the Helper Script

You need to perform the following modifications to the Windows installation (which are beyond the options of Windows Unattended CD Creator):

- Allow remote desktop access to the operating system after its installation on the OSN module.
- Disable the firewall for blocking remote access.

These modifications can be performed by a **commandfile** run at the end of the installation. Use Notepad or a similar text editor to create the following two files and save them in a separate folder named "M1K" on your PC (for later use by the Windows Unattended CD Creator):

- File name: **do.cmd**

```
copy d:\m1k\nofw.cmd "%systemdrive%\Documents and Settings\All
Users\Start Menu\Programs\Startup"
netsh firewall set opmode disable
reg add
"HKEY LOCAL MACHINE\SYSTEM\CurrentControlSet\Control\Terminal
Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f"
reg add
"HKEY LOCAL MACHINE\SYSTEM\CurrentControlSet\Control\Terminal
Server" /f /v TSAdvertise /t REG_DWORD /d 1 /f"
md %systemdrive%\m1k
xcopy d:\m1k %systemdrive%\m1k
exit
```

- File name: **nofw.cmd**

```
netsh firewall set opmode disable
```

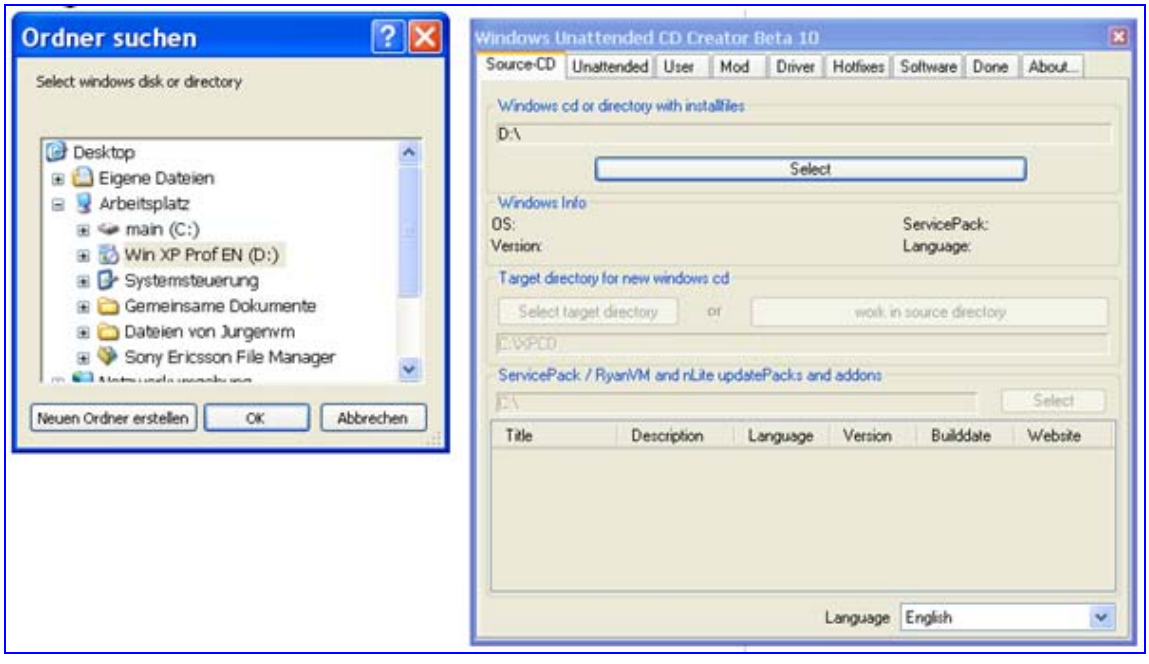
13.2.2 Stage 2: Creating the ISO Image

The procedure below describes how to create the ISO image.

➢ To create the ISO image, take these 13 steps:

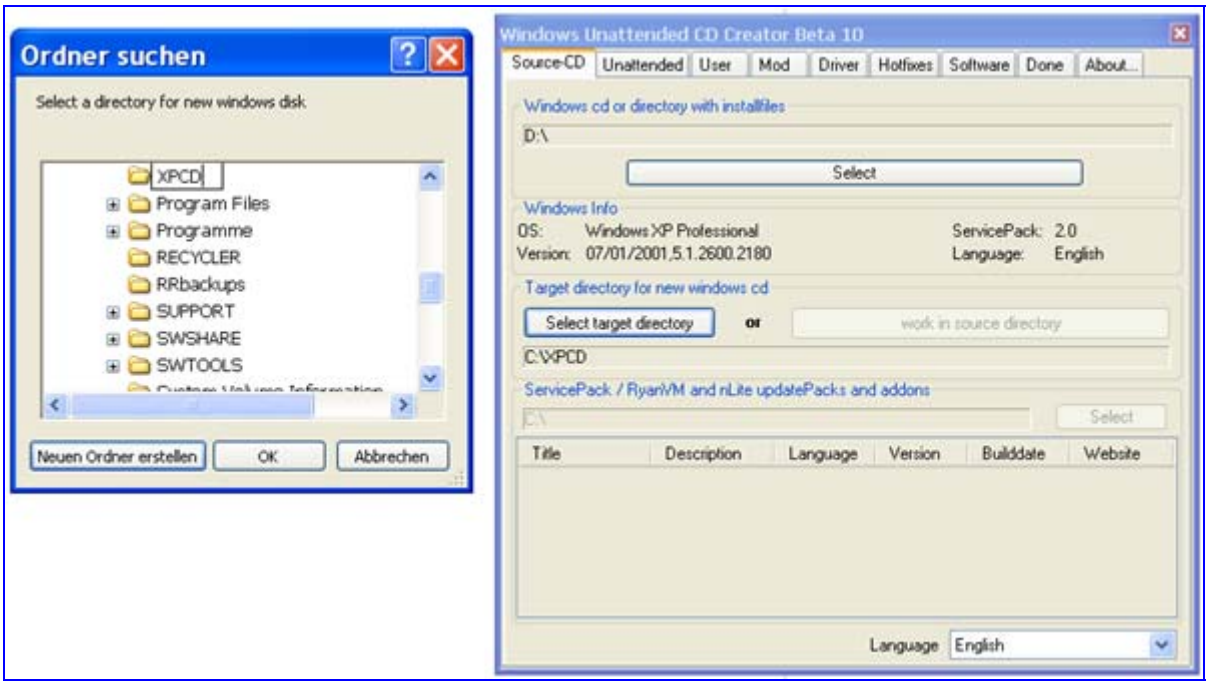
1. Insert the original Windows installation disk in your CD drive, and then close any autorun screen that may appear.
2. Start Windows Unattended CD Creator, and then click **Select** to indicate the source of the original Windows CD.

Figure 13-1: Selecting Source of Windows XP Installation CD



3. Define the target directory for the new Windows CD.

Figure 13-2: Selecting Target Directory for Windows XP

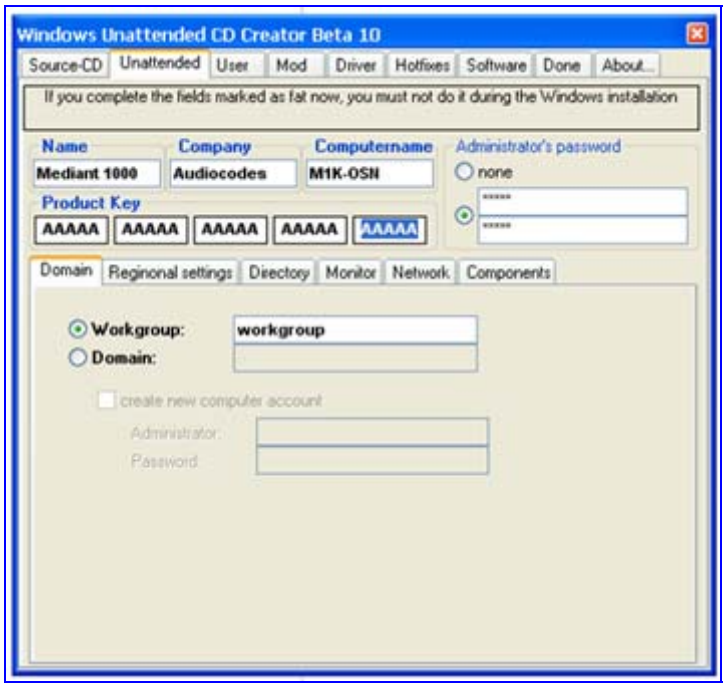


4. Wait for the copy process to complete.

5. Use Windows Explorer to copy the "M1K" folder to the target directory. This is required to allow the .cmd scripts to operate correctly. (Note: you can also copy your own programs and files into this "M1K" folder. They will be copied to the c:\M1K folder on the OSN module as part of the Windows installation.)

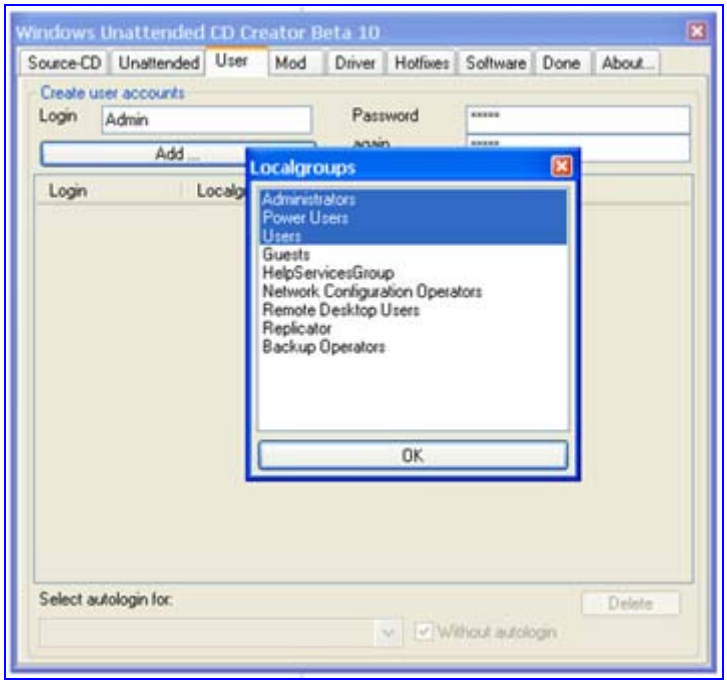
6. Enter all user and registration details and define a password for the administrator (failure to do so will result in invisible prompts during installation).

Figure 13-3: Defining User and Registration Details



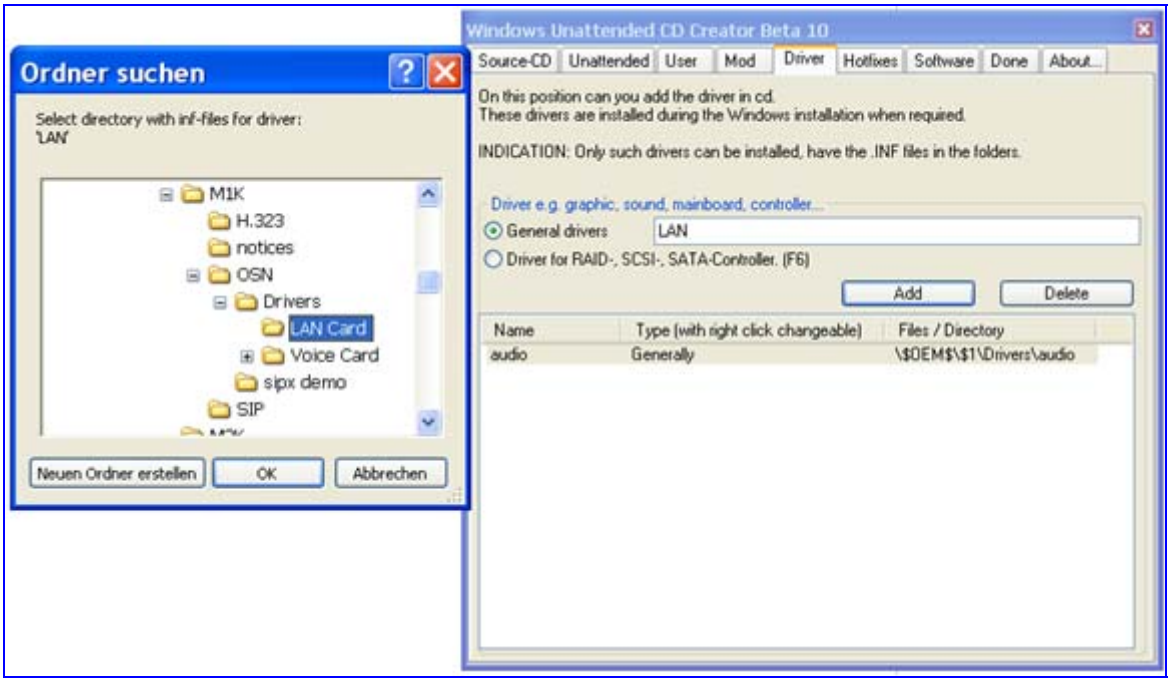
7. Create a default user with a password, and assign it to the Administrators group.

Figure 13-4: Creating a User



8. Add the drivers for LAN and AUDIO.

Figure 13-5: Adding LAN and Audio Drivers

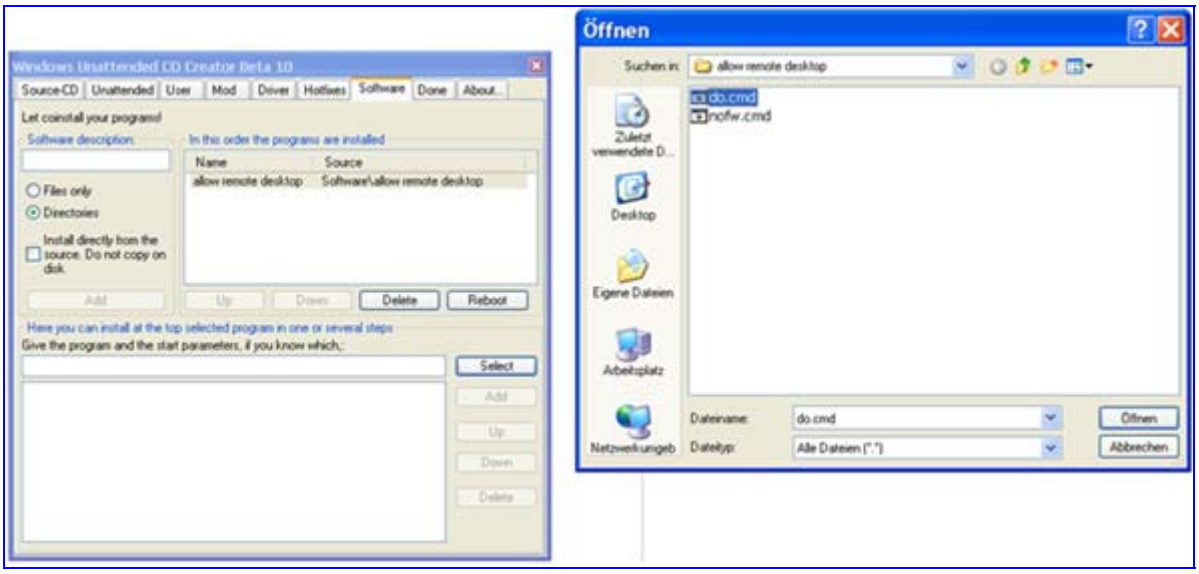


9. Add the directory with the cmd files to the CD. Ensure to select the 'Directories' check box (the description is arbitrary).

Figure 13-6: Adding cmd Files Directory to CD

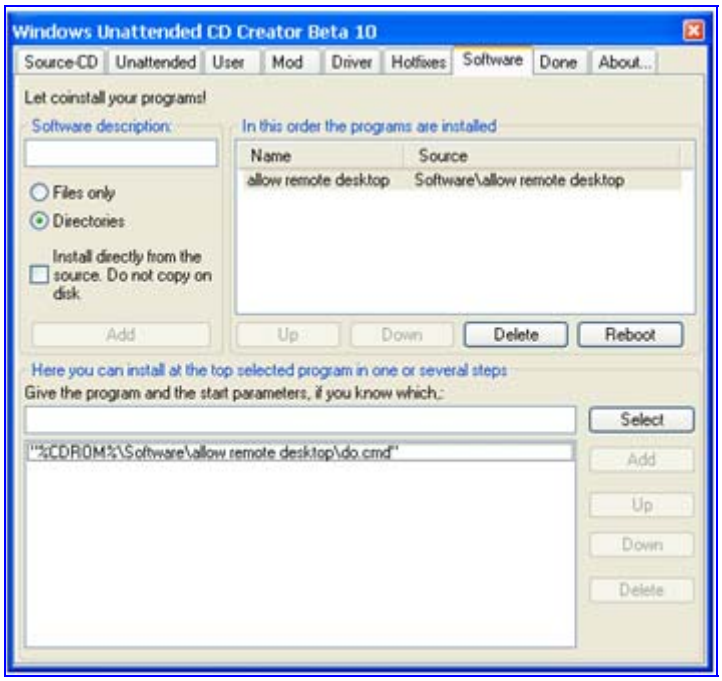
10. Click the newly created name, and then click **Select** to browse to the do.cmd file.

Figure 13-7: Selecting do.cmd File



11. Click **Add** to include this command file to the installation script. The result appears as follows:

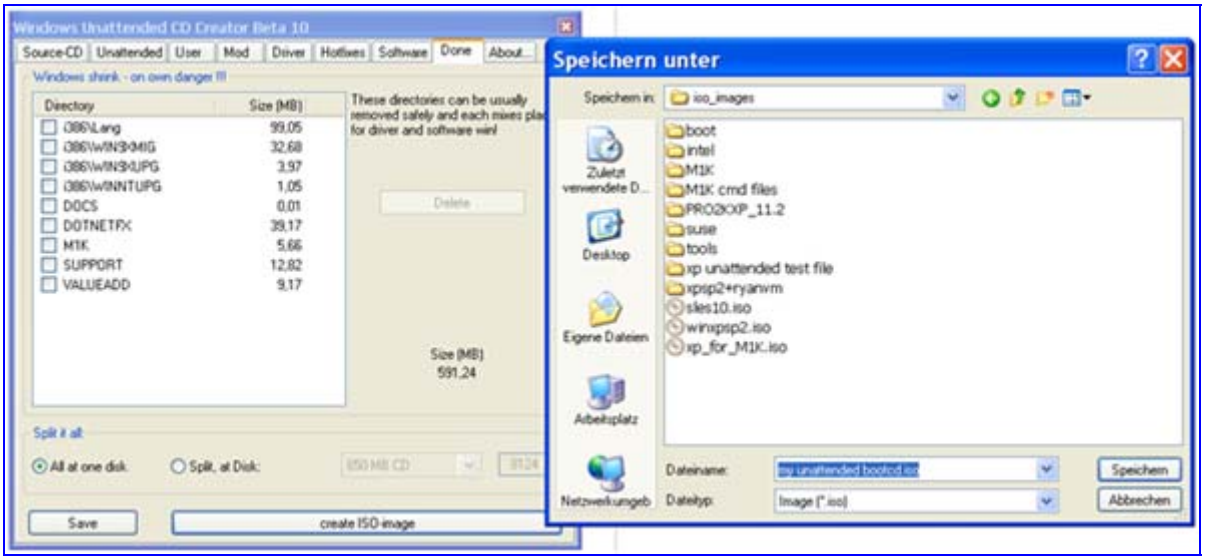
Figure 13-8: Adding cmd File to Installation Script



12. Select the **Done** tab, and then click **Save**.

13. Click **create ISO-image**.

Figure 13-9: Completing ISO Image Creation



Your ISO image is now ready.

13.2.3 Stage 3: Burning the CD

Use a CD burning program to make a CD from the ISO image. Burn with verification to ensure the CD is correctly burned.

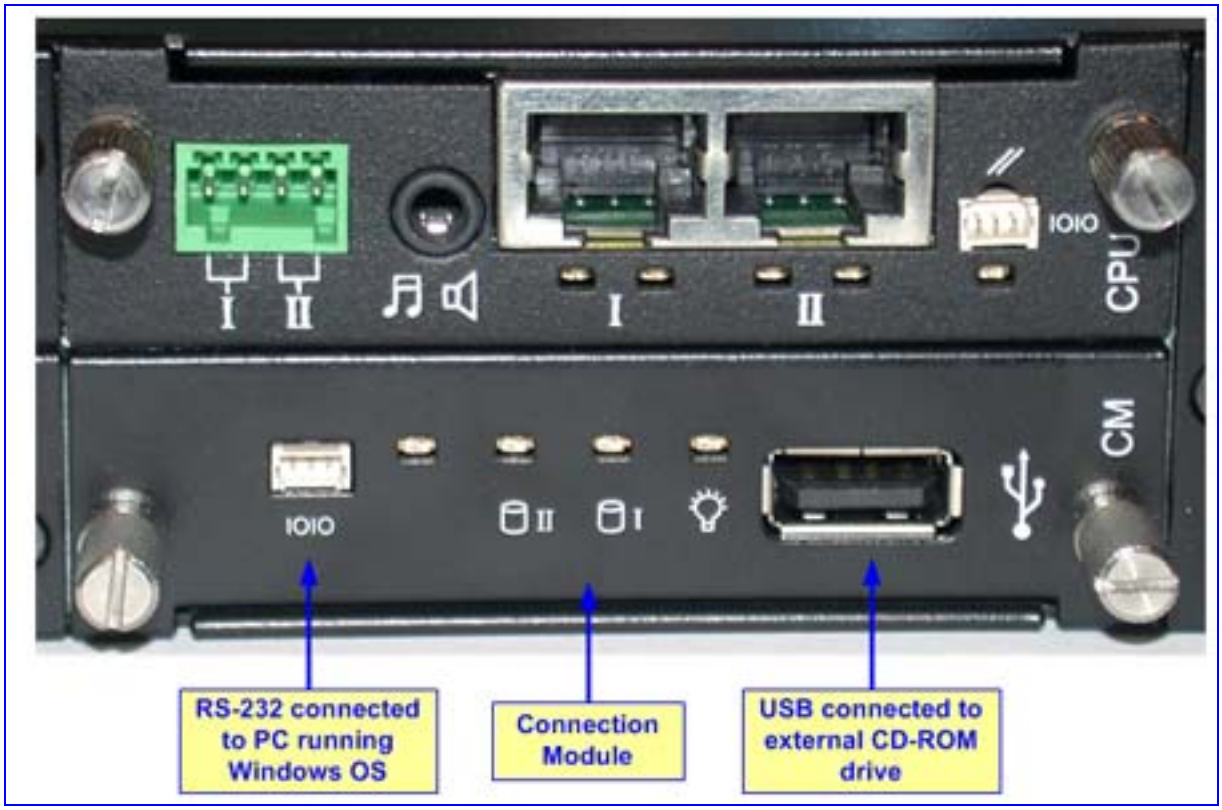
13.2.4 Stage 4: Cabling the Mediant 1000 OSN Server

The procedure below describes the cabling for installing Windows.

➢ To cable the OSN server, take these 3 steps:

1. Connect the CM module's (located on the Mediant 1000 front panel) RS-232 port to a PC (DB-9 connector), using the RS-232 serial cable.
2. Connect the CM module's USB port to a USB hub, using the USB cable.
3. Connect an external CD-ROM drive and a keyboard to the USB hub.

Figure 13-10: Mediant 1000 Front Panel OSN Server Connections



13.2.5 Stage 5: Booting and Installing

Once you have cabled the Mediant 1000 OSN Server (as described in 'Stage 4: Cabling the Mediant 1000 OSN Server' on page 515), you can boot and install Windows XP.

➢ To boot and install, take these 5 steps:

1. Start HyperTerminal on the computer and set the baud rate to 115200, 8N1.
2. Power up Mediant 1000; HyperTerminal displays startup messages. There are three possible scenarios:

3. **Alternative 1:** The CD immediately boots. This indicates that the hard disk is unformatted and unpartitioned.
   a. Allow CD activity and intermediate pauses to run for about 10 minutes.
   b. Once CD activity has stopped, perform the following, using the USB keyboard:
      a. Press the <Enter> key.
      b. To quicken installation, press the <Up> arrow key twice. This changes NTFS full format to NTFS quick format, eliminating time required for hard disk formatting.
      c. Press the <Enter> key; CD activity restarts, with several pauses. The installation runs for at least 60 minutes after which all CD activity and Mediant 1000 hard disk activity ends.
      d. Using Wireshark, check for BootP broadcasts to find the OSN module's IP address. Alternatively, you can use the DHCP server for locating the address that was assigned to the OSN MAC address.
      e. Use Remote Desktop Connection to connect to the OSN module. Use the user name, password, and domain that you specified in 'Stage 2: Creating the ISO Image' on page 510.
4. **Alternative 2:** HyperTerminal displays the message “press any key to boot from CD.”. This indicates that the hard disk contains a valid partition.

Warning: The procedure described here will remove all partition data from the OSN hard disk. CONTINUE ONLY IF YOU ARE SURE YOU NO LONGER NEED ANY INFORMATION FROM THE FIRST PARTITION ON THE DISK.

   a. If you are certain that this is a single partition that you no longer need, press the <Enter> key to continue and boot from the Windows installation disk.
   b. Allow CD activity and intermediate pauses to run for about 10 minutes.
   c. Once CD activity has stopped, perform the following, using the USB keyboard:
      a. Press the ‘D’ key to clear the first partition.
      b. Press the <Enter> key.
      c. Press the 'L' key to confirm removal of the partition.
      d. To quicken installation, press the <Up> arrow key twice. This changes NTFS full format to NTFS quick format, eliminating time required for hard disk formatting.
      e. Press the <Enter> key; CD activity restarts, with several pauses. The installation runs for at least 60 minutes after which all CD activity and Mediant 1000 hard disk activity ends.
      f. Using Wireshark, check for BootP broadcasts to find the OSN module's IP address. Alternatively, you can use the DHCP server for locating the address that was assigned to the OSN MAC address.
      g. Use Remote Desktop Connection to connect to the OSN module. Use the user name, password, and domain that you specified in 'Stage 2: Creating the ISO Image' on page 510.
5. **Alternative 3:** The boot process stops after “press any key to boot from CD….” message. This may indicate that the OSN disk currently contains a Linux installation. Use a Linux recovery BOOT-CD or other mechanisms to remove all partition information, and then retry the Windows Boot CD. For example, use the Suse Linux CD where the “label rescue” in isolinux.cfg was modified to run with serial console.
   a. Type **fdisk /dev/hda** to start the partition manager.
   b. Type **d** and **w** to clear and rewrite the partition table.

13.2.6 Stage 6: Enabling the Firewall with Remote Desktop Access

The OSN server's firewall was initially disabled to allow Windows XP installation and Remote Desktop access (in Stage 1: Creating the Helper Script). However, once you access the OSN Server after Windows installation has complete, you need to enable the Windows XP firewall and allow future Remote Desktop access.

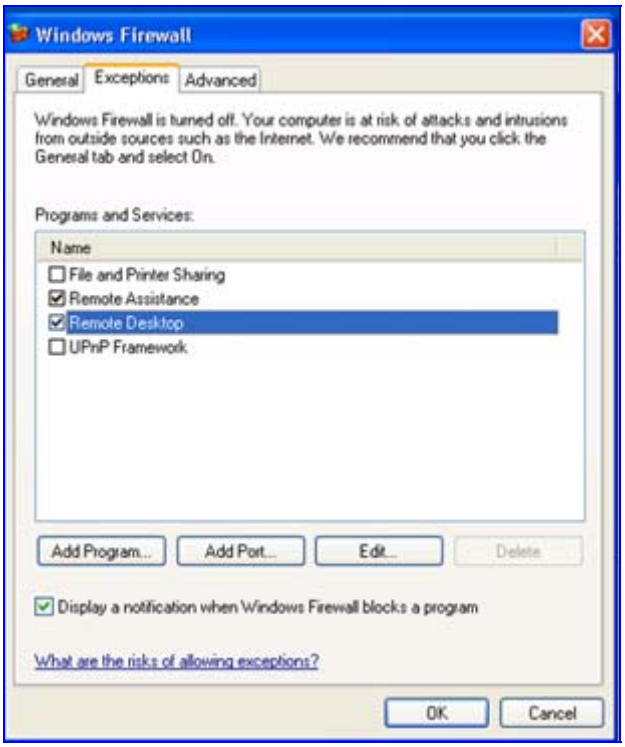
➢ **To enable the Windows XP firewall and allow Remote Desktop access, take these 4 steps:**

1. Open Windows Security Center (from **Start** menu, point to **Settings**, **Control Panel**, and then click **Security Center**).

Figure 13-11: Windows Security Center

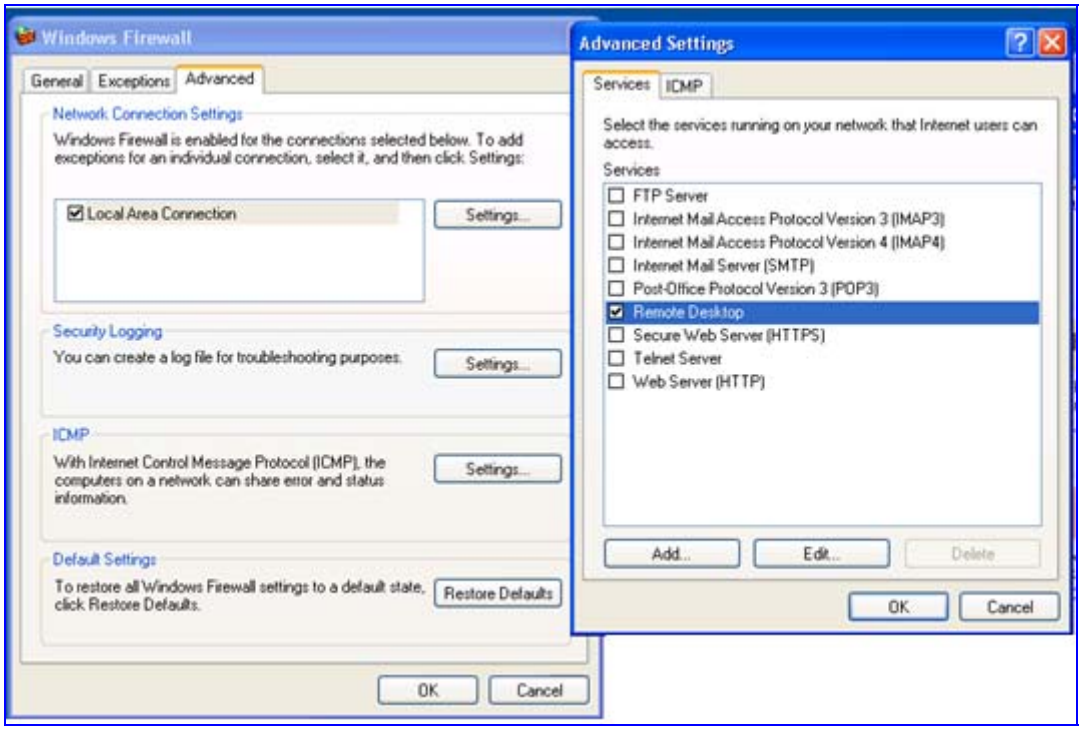
2. Click **Windows Firewall**, and then add 'Remote Desktop' to the selected firewall exceptions.

Figure 13-12: Remote Desktop Firewall



3. Select the **Advanced** tab, click **Settings** for the 'Local Area Connection', and then also allow Internet access to Remote Desktop.

Figure 13-13: Allowing Internet Users Access to Remote Desktop



4. Delete the 'nofw.cmd' item from Windows **Startup** menu.

13.3 Connecting Remotely to OSN Server using Windows XP

Typically, for customers requiring Microsoft Windows® XP operating system (OS), the OSN Server is provided with Windows XP already pre-installed. Once the OSN Server modules have been installed in the Mediant 1000 chassis, you can connect to the OSN Server using Microsoft's Remote Desktop Connection program.

Notes:

- This section is applicable only to Mediant 1000.
- For installing Linux OS, refer to "Installing Linux™ Operating System on the OSN Server" on page 523.

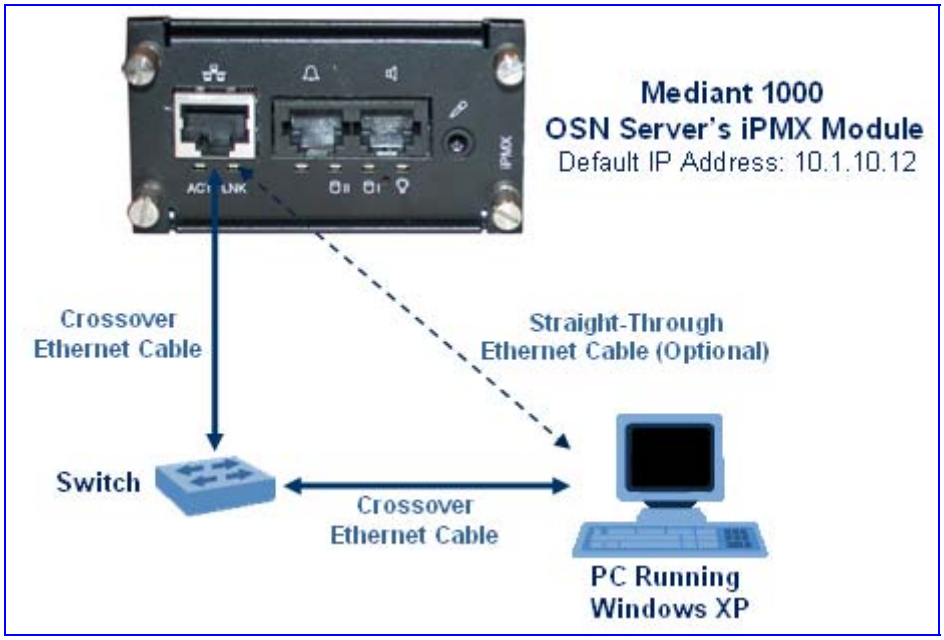
13.3.1 Cabling

Before you can connect remotely to the OSN Server, you need to cable a PC (running Remote Desktop Connection) to the OSN Server's iPMX module.

➢ To cable a PC to the OSN Server running Windows XP for remote connection, take the following step:

- Connect a PC to the Ethernet LAN port on the Mediant 1000 iPMX module (located on the rear panel), by performing one of the following:
  - Using a crossover cable, connect the PC's LAN port directly to the iPMX LAN port.
  - Using a straight-through cable, connect the PC's LAN port to a switch. Connect the switch to the iPMX LAN port.

Figure 13-14: Cabling iPMX for Remote Connection from PC with Windows XP



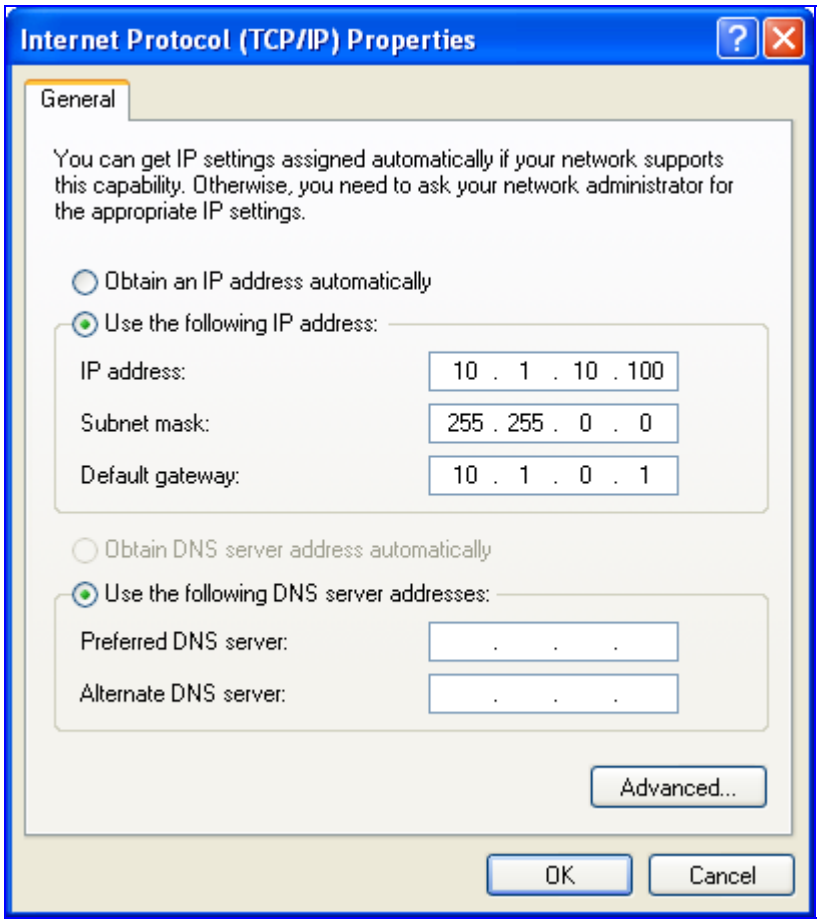
13.3.2 Connecting Using Remote Desktop Connection

Once you have cabled the PC to the OSN Server's iPMX module, perform the procedure below for connecting the PC remotely to the OSN Server (running Windows XP) using the Remote Desktop Connection program.

➢ To remotely connect a PC to the OSN Server running Windows XP, take these 6 steps:

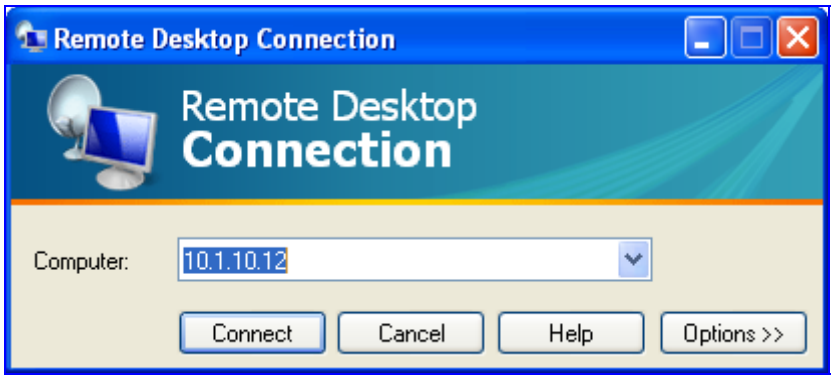
1. Change the PC's IP address so that it is in the same subnet as the default OSN Server's IP address (i.e., 10.1.10.12). The figure below displays an example of a changing a PC's IP address:

Figure 13-15: Changing the PC's IP Address


Internet Protocol (TCP/IP) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address: 10 . 1 . 10 . 100
Subnet mask: 255 . 255 . 0 . 0
Default gateway: 10 . 1 . 0 . 1
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:
Advanced...
OK
Cancel

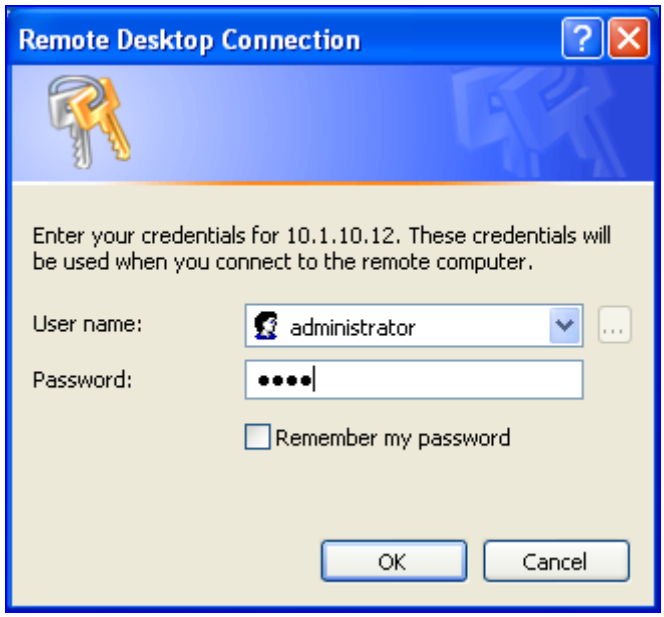
2. Start Microsoft's Remote Desktop Connection program - from the **Start** menu, point to **Programs**, to **Accessories**, to **Communications**, and then click **Remote Desktop Connection**.

Figure 13-16: Entering IP Address in Remote Desktop Connection



3. In the 'Computer' field, enter the OSN Server's default IP address (i.e., 10.1.10.12).
4. Click **Connect**.

Figure 13-17: Entering User Name and Password in Remote Desktop Connection



5. Enter the OSN Server's default user name (i.e., "administrator") and password (i.e., "1234").
6. Click **OK**; Remote Desktop Connection connects you to the desktop of the Mediant 1000 gateway's OSN server.

Reader's Notes

14 Installing Linux™ Operating System on the OSN Server

Once the OSN Server modules have been installed in the Mediant 1000 Media Gateway, you need to install an operating system (OS) on the OSN server on which the partner application (e.g. IP-PBX) is to run. If you require the Linux™ OS, follow the procedures described in this section for installing one of the following distributions of the Linux™ OS on the Mediant 1000 OSN server on which the partner application is to run:

- Linux™ RedHat (and Fedora)
- Linux™ Debian
- Linux™ SUSE

Notes:

- This section is applicable only to Mediant 1000.
- For OSN server running Windows XP OS, refer to Connecting Remotely to OSN Server using Windows XP on page 509.

This section can also serve as a reference for installing other Linux™ distributions. These instructions have been verified against the following distributions:

- RedHat™ Linux 9
- RedHat™ Fedora Core 4
- RedHat™ Enterprise 3
- RedHat™ Enterprise 4
- Debian™ 3.1 (r0a) “Sarge”
- SUSE™ 9.3
- SUSE™ 10

Note: Redhat™ Fedora Core 3 is not supported.

14.1 Requirements

The following subsections list the hardware and software requirements for installing the Linux™ operating system on the Mediant 1000 OSN server.

14.1.1 Hardware

Before installing Linux on the Mediant 1000 OSN Server, ensure you have the following:

- Ethernet cable cord
- External USB CD ROM or DVD ROM (not supplied)

- USB cable (not supplied) to connect the external USB CD ROM to the Mediant 1000
- RS-232 cable (supplied)
- Linux™ Distributions Installation CDs
- Blank CD or DVD media
- Windows™ PC with CD RW or external CD ROM/RW

14.1.2 Software

The software requirements include the following:

- RS-232 console / terminal software (e.g., HyperTerminal™)
- ISO Image Editor (WinISO™ is recommended -- refer to http://www.winiso.com/download.htmUNIX™ File Format text editor (refer to http://www.pspad.comr http://www.ultraedit.com

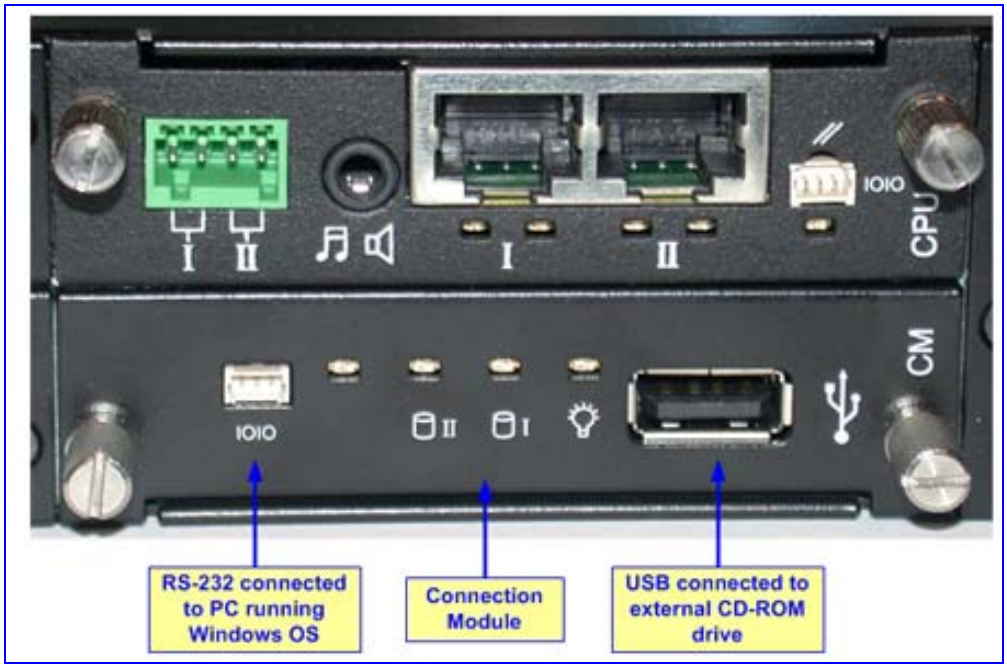
14.2 Cabling

The following procedure describes the cabling before installing Linux™ operating system on the Mediant 1000 OSN Server.

➢ **To cable the OSN Server for installing the Linux™ operating system, take these 4 steps:**

1. On the Mediant 1000 CM module (located on the front panel), perform the following:
   a. Connect the RS-232 port to a PC, using the RS-232 cable.
   b. Connect the USB port to an external CD-ROM drive, using the USB cable.

Figure 14-1: Mediant 1000 Front Panel OSN Server Connections



2. On the Mediant 1000 iPMX module (located on the rear panel), connect the RJ-45 Ethernet port, using the Ethernet cable.

3. Connect the external CD-ROM to the power supply.
4. Connect the Mediant 1000 to the power supply.

14.3 Installing Linux™ RedHat (and Fedora)

Perform the following four stages for installing Linux™ Redhat (and Fedora). (Some distributions of Linux™ may vary slightly):

Notes:

- Some distributions of Linux may vary slightly.
- The Linux version for installation must be according to the application requirements (e.g., Asterisk requires RHAT ES Ver. 3.0; Pingtel ECS requires RHAT ES Ver. 4.0).

14.3.1 Stage 1: Obtaining the Linux Redhat ISO Image

To obtain an updated ISO image, perform one of the following:

- Download it from the AudioCodes Web site, as described in "Downloading an updated LinuxTM Redhat ISO Image" on page 525,
- Create it using the steps detailed in "Creating an updated Linux Redhat ISO Image" on page 526.

14.3.1.1 Downloading an Updated ISO Image

➢ **To download an ISO image from AudioCodes' Web site, take these 6 steps:**

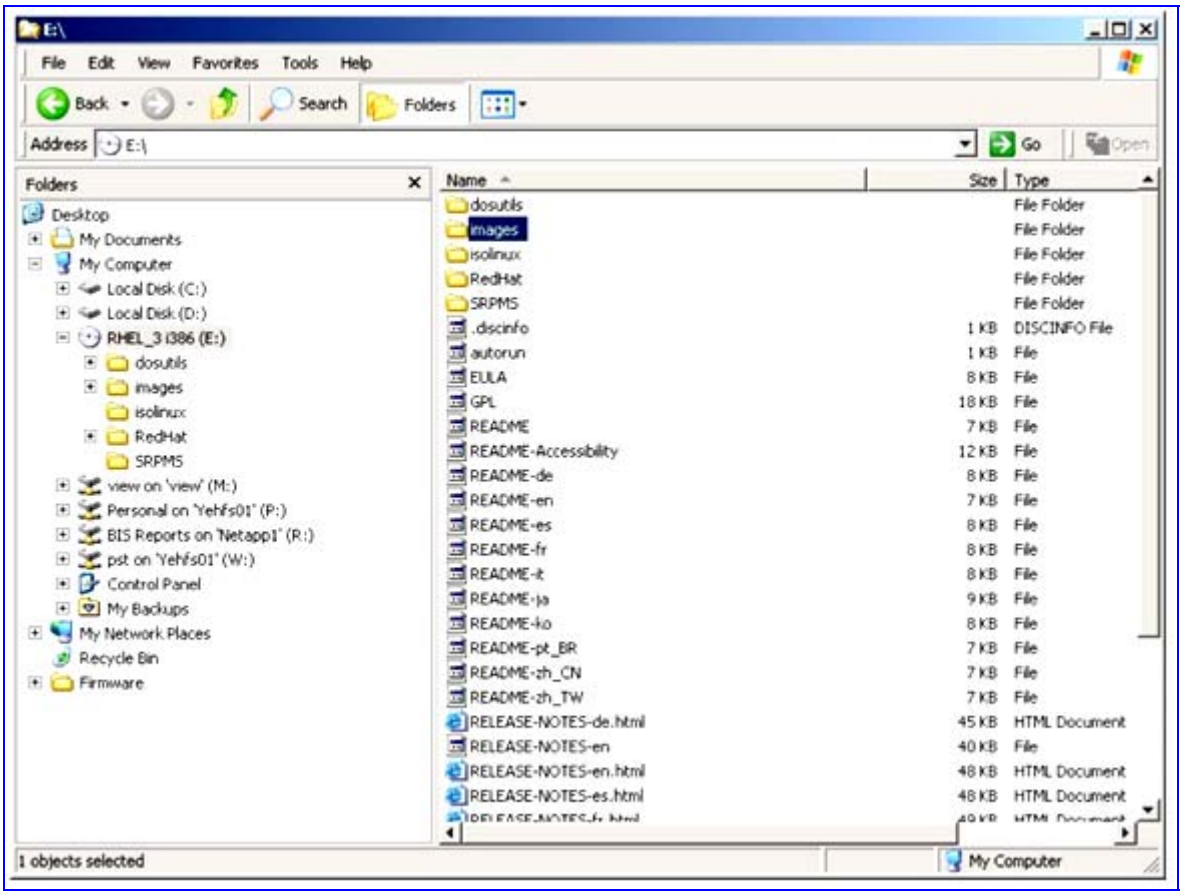
1. Access AudioCodes' Web site (http://www.audiocodes.com), and then navigate to the 'Support' page.
2. Click the **Registered Users Login** link, and then login with your username and password.
3. Under 'Product Documentation', select the **Mediant 1000** link, and then click **Mediant 1000 OSN Server**.
4. Select **Linux Boot Image**, and then select the required installation.
5. Download the ISO image to a folder called 'Partner Install' on your PC.
6. Continue with Stage 3 to burn the CD ("Stage 3: Burn the CD" on page 532).

14.3.1.2 Creating an Updated ISO Image

➢ **To create an updated Linux Redhat ISO Image take these 6 steps**

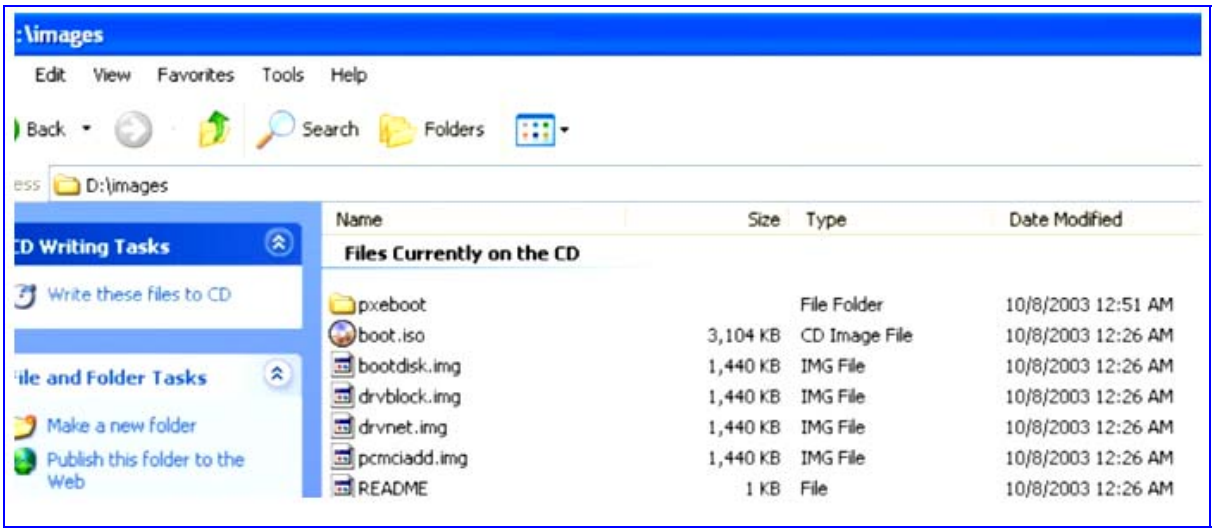
1. On the local hard disk of the Window's™ PC, create a new folder called 'Partner Install'.
2. If you have not already done so, download a utility that allows editing of an ISO image (e.g., WinISO™ from http://www.winiso.com/download.htm
3. Using Internet Explorer, download a UNIX File Format text editor (e.g., PSPad™ at http://www.pspad.comr UltraEdit™ at http://www.ultraedit.com
4. Insert the first installation disk of the Linux™ Redhat distribution into the CD-ROM drive of the Windows™ PC. The Windows Explorer screen appears, displaying files currently on the CD.

Figure 14-2: Disk 1 of Redhat Partner Installation



5. Locate the *boot.iso* file in the images folder on the CD (refer to the note below).

Figure 14-3: Images Folder


:\images
Edit View Favorites Tools Help
Back Search Folders
D:\images
CD Writing Tasks
Write these files to CD
File and Folder Tasks
Make a new folder
Publish this folder to the Web
Name Size Type Date Modified
Files Currently on the CD
pxeboot File Folder 10/8/2003 12:51 AM
boot.iso 3,104 KB CD Image File 10/8/2003 12:26 AM
bootdisk.img 1,440 KB IMG File 10/8/2003 12:26 AM
drvblock.img 1,440 KB IMG File 10/8/2003 12:26 AM
drvnet.img 1,440 KB IMG File 10/8/2003 12:26 AM
pcmciadd.img 1,440 KB IMG File 10/8/2003 12:26 AM
README 1 KB File 10/8/2003 12:26 AM

6. Copy the *boot.iso* file to the 'Partner Install' folder created in Step 1, and then open it with an ISO image editor.

   The *isolinux.cfg* file should appear as shown in the screen below.

Figure 14-4: ISO Screen

WinISO (Buy now! Only USD30) - C:\Partner Install\boot.iso (Unregistered Version)
File Actions Bootable CD Convert Options Help
New Open Save Add Extract View Convert Buy Now!
Current Path: \isolinux
Filename Type Size Time
boot.cat Security Catalog 2048 2003-10-07 1...
boot.msg Outlook Item 364 2003-10-07 1...
general.msg Outlook Item 1026 2003-10-07 1...
initrd.img IMG File 2148969 2003-10-07 1...
isolinux.bin BIN File 9428 2003-10-07 1...
isolinux.cfg CFG File 435 2003-10-07 1...
options.msg Outlook Item 660 2003-10-07 1...
param.msg Outlook Item 869 2003-10-07 1...
rescue.msg Outlook Item 557 2003-10-07 1...
snake.msg Outlook Item 549 2003-10-07 1...
splash.lss LSS File 6056 2003-10-07 1...
vmlinuz File 888750 2003-10-07 1...
Bootable CD
Total 2 MB 0% Max. 700MB
1 files selected, 435 Bytes 12 files, 2 MB

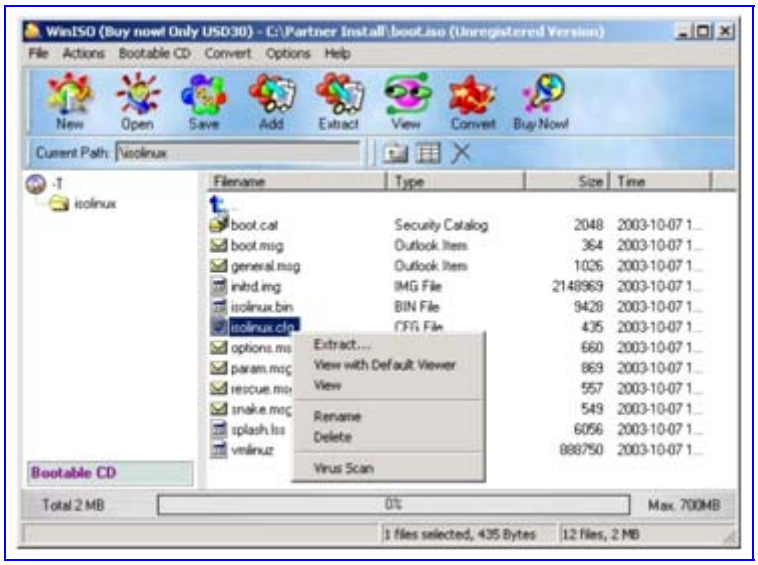
Note: The 'images' folder may be named differently on different Linux™ distributions.

14.3.2 Stage 2: Editing the isolinux.cfg File

➢ To edit the isolinux.cfg file, take these 14 steps:

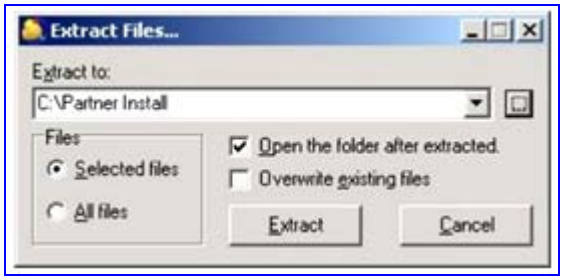
1. Extract the *isolinux.cfg* file by performing the following:
   a. Right-click the *isolinux.cfg* file, and then from the shortcut menu, choose **Extract**.

Figure 14-5: Selecting Extract Option


WinISO (Buy now! Only USD30) - C:\Partner Install\boot.iso (Unregistered Version)
File Actions Bootable CD Convert Options Help
New Open Save Add Extract View Convert Buy Now!
Current Path: \isolinux
Filename Type Size Time
isolinux
boot.cat Security Catalog 2048 2003-10-07 1...
boot.msg Outlook Item 364 2003-10-07 1...
general.msg Outlook Item 1026 2003-10-07 1...
initrd.img IMG File 2148969 2003-10-07 1...
isolinux.bin BIN File 9428 2003-10-07 1...
isolinux.cfg CFG File 435 2003-10-07 1...
Extract...
View with Default Viewer
View
Rename
Delete
Virus Scan
Bootable CD
Total 2 MB
0%
Max. 700MB
1 files selected, 435 Bytes
12 files, 2 MB

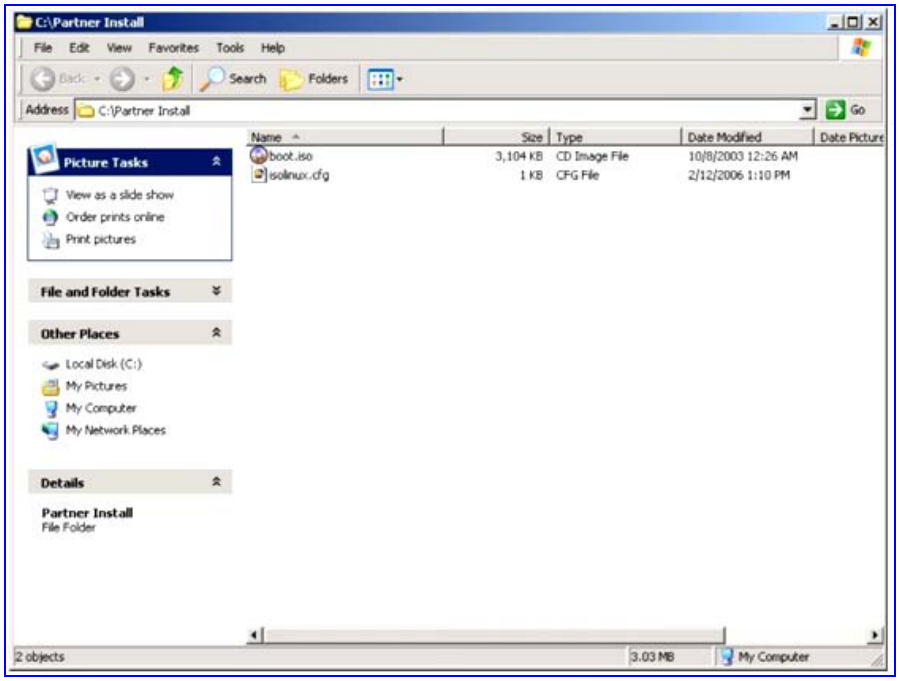
   b. In the 'Extract to' field, browse to the 'Partner Install' folder (created in Stage 1) to where the *isolinux.cfg* file must be extracted.

Figure 14-6: Extracting Files to Partner Install Folder



c. Click **Extract**; the files is extracted and a screen opens containing the extracted *isolinux* file.

Figure 14-7: ISO-Extract Screen



2. Open the *isolinux.cfg* file with a text editor that supports UNIX file format (e.g., PSPad or UltraEdit); the following screen appears.

Figure 14-8: Text Edit Screen

```
default linux
prompt 1
timeout 600
display boot.msg
F1 boot.msg
F2 options.msg
F3 general.msg
F4 param.msg
F5 rescue.msg
F7 snake.msg
label linux
  kernel vmlinuz
  append initrd=initrd.img
label text
  kernel vmlinuz
  append initrd=initrd.img text
label expert
  kernel vmlinuz
  append expert initrd=initrd.img
label ks
  kernel vmlinuz
  append ks initrd=initrd.img
label lowres
  kernel vmlinuz
  append initrd=initrd.img lowres
```

3. Insert the following line at the beginning of the file so that it's the first line:

```
serial 0 115200
```

4. Locate the line 'default <my_label>' (usually 'default linux' appears), and then locate the line 'label <my_label>' (usually 'label linux' appears). Under this line, the following appears:

```
kernel ...
append ...
```

5. Add the following parameters to the 'append' line of <my_label>:

```
text console=ttyS0,115200
```

Note: In the above string, “ttyS0,115200” consists of a capital “S”, only zeros, and one comma.

The 'kerne' and 'append' lines should now look like the following example:

```
label linux
kernel vmlinuz
append initrd=initrd.img ramdisk_size=8192
text console=ttyS0,115200
```

6. Locate the line 'prompt <flag>' (usually 'prompt 1' appears) and change it to 'prompt 0'.

7. Locate the line 'timeout <tenth_of_secs>' (usually 'timeout 600' appears) and change it to 'timeout 0'.

Note: If the timeout line does not exist, **do not** add it.

The *isolinux.cfg* file should now look like the following:

```
serial 0 115200
default linux
prompt 0
...
label linux
kernel vmlinuz
append initrd=initrd.img ramdisk_size=8192
text console=ttyS0,115200
```

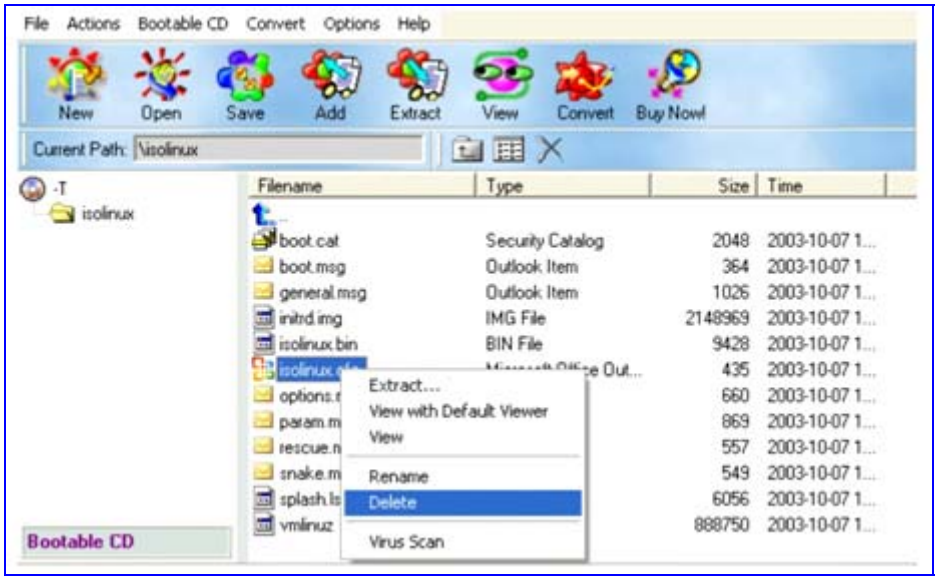
8. Save the changes to the *isolinux.cfg* file, and then close the text editor.

9. Navigate to the 'Partner Install' folder and with the ISO editing utility, open the *boot.iso* file.

10. Double-click the 'isolinux' directory; the folder's contents is displayed.

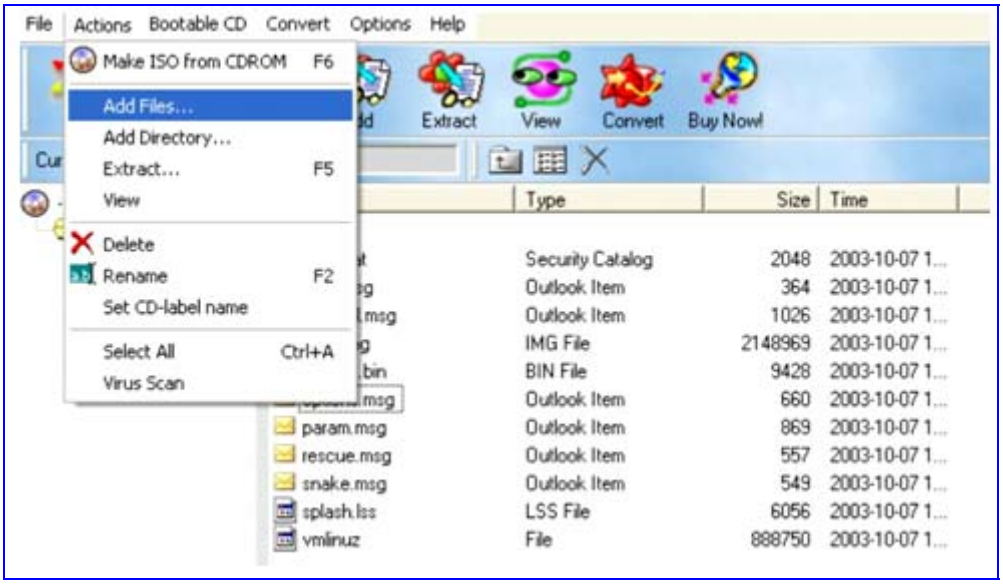
11. Right-click the *isolinux.cfg* file, and then from the shortcut menu, choose **Delete** to delete this file.

Figure 14-9: Deleting CFG



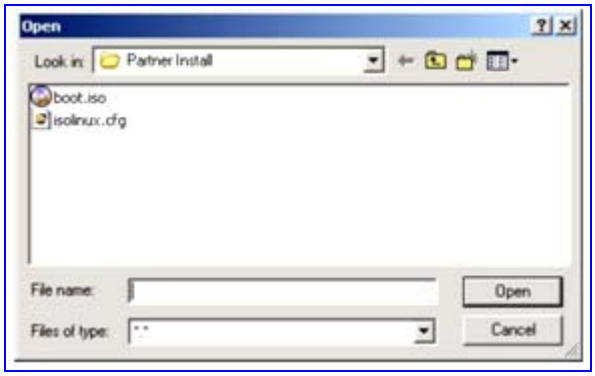
12. From the ISO edit utility menu, select the **Actions** option, followed by **Add Files**.

Figure 14-10: File Add



13. Navigate to the 'Partner Install' folder, select the *isolinux.cfg* file, and then click **Open**.

Figure 14-11: ISO Open Function



The updated *isolinux.cfg* file has now been copied from the 'Partner Install' folder to the *boot.iso* image.

14. Save the *boot.iso* image in the 'Partner Install' folder.

14.3.3 Stage 3: Burning ISO Image File to CD-ROM

➢ To burn the boot.iso file to a CD-ROM, take these 3 steps:

1. Open a burning utility.
2. Use the **Burn Image** option to burn the *boot.iso* file to an empty CD media.
3. Label the media as “Boot CD”.

Note: Ensure that the *boot.iso* file is burned to the CD as an image and not as a data file.

14.3.4 Stage 4: Installing the Boot Media

Now you have the boot media which enables the installation of the Mediant 1000 using the serial connection (terminal) with RS-232 cable.

Note: Some third-party applications require specific Linux OS installation steps. Once you have completed the basic installation procedure explained in this section, refer to the relevant third-party application user's manual for a detailed explanation on installing the Linux OS per third-party application.

➢ To complete the installation, take these 9 steps:

1. Connect your Windows™ PC to the Mediant 1000 using a serial cable.
2. Connect the USB CD-ROM device to the Mediant 1000 using a USB cable.
3. Power up the PC.

4. Open the Terminal application (e.g. HyperTerminal) on your Windows™ PC. Create a new connection with the following settings:
   - Connect Port: COM1
   - Baudrate: 115200 (bits per second)
   - Data Bits: 8
   - Parity: None
   - Stop Bits: 1
   - Flow Control: None
5. Insert the "Boot CD" (created in Stage 3) into the USB CD-ROM drive.
6. Power up the Mediant 1000.
7. On the Terminal application, the BIOS phase starts and the Linux installation begins. The installation uncompresses the kernel, loads it and its drivers, and then starts the interactive installation.

Note: After the BIOS phase, some badly formatted text may appear on the screen.

8. The first interactive screen should be 'Choose a Language'. Select the language you wish to use.

Figure 14-12: Choose a Language

9. When prompted for the installation media, **remove the "Boot CD" and insert the first installation disk** and select 'CDROM' as your installation media. From this point on, you should proceed with the screen instructions, as instructed by your Linux™ distributor.

14.3.5 Additional RedHat™ and Fedora™ Installation Notes

Please refer to the following notes for the remaining part of the installation.

1. Select **LILO** as your bootloader where possible, otherwise select GRUB.
2. It is recommended that you disable the firewall when prompted (select "**No Firewall**").
3. If you forget to disable the firewall during the installation and want to do it after the installation, run the following command:
   ```
   /usr/bin/redhat-config-securitylevel-tui --quiet –disabled
   ```
4. It is recommended that you assign a **static IP address** to your Mediant 1000. So when the installation has been completed, you will be able to create an SSH remote connection and continue the post-installation configuration.
5. During the bootloader configuration (after you selected which bootloader you want to install), you will be prompted to provide additional text to be appended to the kernel. Ensure that the installation (grub-installer or lilo-installer) recognizes the serial console and contains the following text:
   ```
   console=ttyS0,115200
   ```
6. If there is no text (i.e. the installer did not recognize the serial console), then insert the following:
   ```
   text console=ttyS0,115200
   ```
7. Once the installation is complete, you are prompted to re-start the Mediant 1000.

Note: If the installed kernel is version 2.6 or later (Fedora Core 4 or later, RedHat Enterprise 4 or later) then refer to the Post-installation Notes for Kernels 2.6+, below.

14.3.6 Post-installation Notes for Kernels 2.6+ (Fedora™ Core 4+ and RedHat™ EL 4+)

1. When the Mediant 1000 is re-booting and after the BIOS phase, there is a bootloader phase (GRUB or LILO) which starts uncompressing and loading the kernel. After the kernel is loaded the services will start. While the services are loading and the message "**Press 'I' to enter interactive startup**" appears, press "I".
2. Once the hardware has been detected, you will be prompted whether to start each service. For the **syslog service**, select "N" (in order NOT to load it).
3. After you login (either serially or using SSH) to the Mediant 1000, you should disable the syslog service from being started during system startup. To do this, you have to disable the **S**syslog** file located in both /etc/rc3.d and in /etc/rc5.d directories. The file is usually called **S12syslog**.
4. Rename it to K12syslog with the following command.
   ```
   mv S12syslog K12syslog
   ```

14.4 Installing Linux™ Debian

Perform the following five stages for installing Linux™ Debian.

Note: Some distributions of Linux may vary slightly.

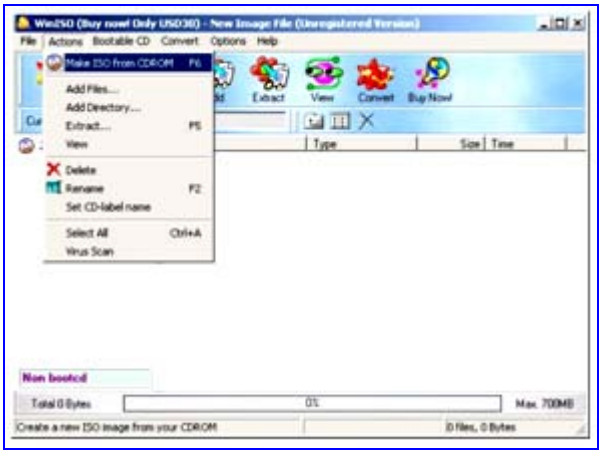
14.4.1 Stage 1: Obtaining the ISO Image

To obtain an updated ISO image, create it using the steps detailed in the section below.

➢ To create an ISO image using an ISO editor utility, take these 4 steps:

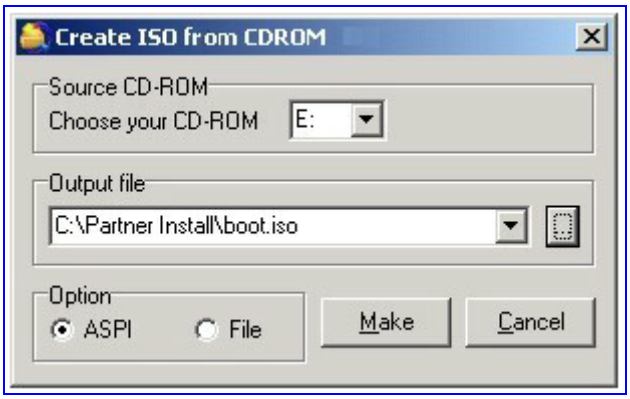
1. Insert the first installation disk of the Linux™ Debian distribution into the CD-ROM drive of the Windows™ PC.
2. Using Internet Explorer, download a utility that allows editing of an ISO image (e.g., WinISO™ at http://www.winiso.com/download.htm).
3. Start WinISO™, and then from the Actions menu, choose **Make ISO from CDROM**.

Figure 14-13: WinISO - Actions Screen



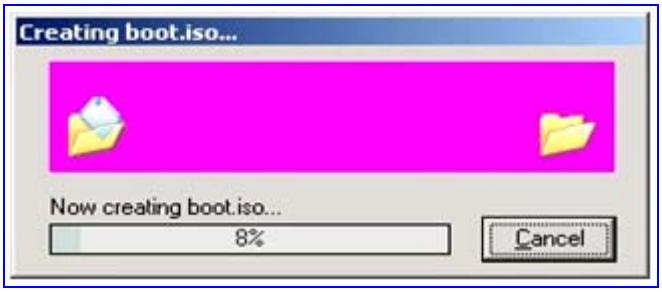
4. Create a 'Partner Install' folder on your hard drive. Select **boot.iso** as the output filename, and then click **Make**.

Figure 14-14: Create ISO from CD-ROM



The *.iso* file starts being created.

Figure 14-15: Creating .iso File



14.4.2 Stage 2: Preparing the Boot Media

➢ To prepare the Boot Media, take these 5 steps:

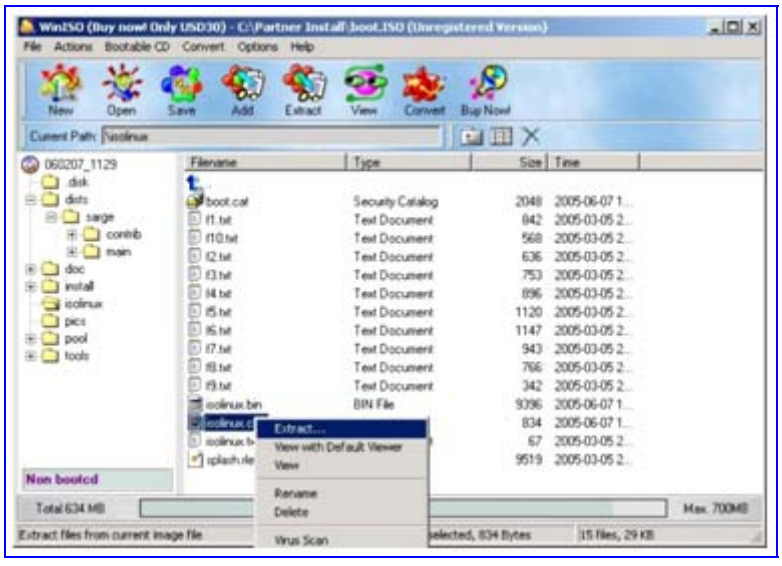
1. If you have not already done so, download a utility that allows editing of an ISO image (e.g., WinISO™ at http://www.winiso.com/download.htm
2. Using Internet Explorer, download a UNIX File Format text editor (e.g., PSPad™ at http://www.pspad.comr UltraEdit™ at http://www.ultraedit.com

3. Locate the *boot.iso* file in the 'Partner Install' folder on the hard disk of your PC and with the ISO image utility, navigate to the \isolinux\isolinux.cfg file.

Figure 14-16: Partner Install Folder

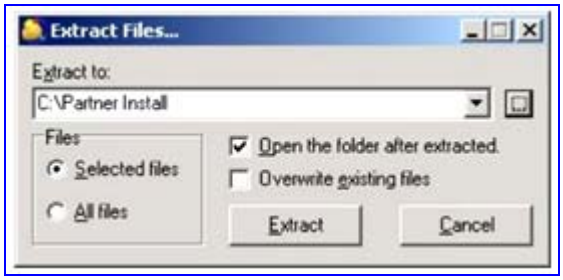
4. Extract the *isolinux.cfg* file by right-clicking the file name, and then from the shortcut menu, choosing **Extract**.

Figure 14-17: Extract isolinux.cfg



5. Extract the *isolinux.cfg* file to the 'Partner Install' folder.

Figure 14-18: Extracting Files to Partner Install Folder



14.4.3 Stage 3: Editing the isolinux.cfg File

To obtain an updated *isolinux.cfg* file, perform one of the following:

- Download it from the AudioCodes Web site as described in "Downloading an updated Debian isolinux.cfg file" on page 538
- Edit it using the steps detailed in "Editing the isolinux.cfg File" on page 538

14.4.3.1 Downloading an Updated Debian isolinux.cfg File

➢ To download an updated Debian isolinux.cfg file from AudioCodes Web site, take these 6 steps:

1. Access AudioCodes' Web site (http://www.audiocodes.com), and then navigate to the 'Support' page.
2. Click the **Registered Users Login** link, and then login with your username and password.
3. Under 'Product Documentation', select the **Mediant 1000** link, and then click **Mediant 1000 OSN Server**.
4. Select **Linux Boot Image**, and then select the required installation.
5. Download the *isolinux.cfg* compressed file to a folder called 'Partner Install' on your PC, and then extract it.
6. Continue with "Editing the isolinux.cfg File" on page 538.

14.4.3.2 Editing the isolinux.cfg File

➢ To edit the isolinux.cfg file, take these 12 steps:

1. Open the 'Partner Install' folder and select the *isolinux.cfg* file with a text editor that supports UNIX file format (e.g., PSPad or UltraEdit).
2. Insert the following line in the beginning of the file, so that it is the first line:
   ```
   serial 0 115200
   ```
3. Locate the line 'default <my_kernel>'. It is usually the first line of the file and appears as follows:
   ```
   DEFAULT /install/vmlinuz
   ```
   Add the following parameters to the 'append' line (located under the 'default' line):
   ```
   text console=ttyS0,115200
   ```

Note: In the above string, “ttyS0,115200” consists of a capital “S”, only zeros, and one comma.

The 'default <my_kernel>' and 'append' lines should look like the following example:

```
DEFAULT /install/vmlinuz
APPEND vga=normal initrd=/install/initrd.gz text
console=ttyS0,115200 ramdisk_size=10240 root=/dev/rd/0
devfs=mount,dall rw -
```

4. Locate the line 'prompt <flag>' (usually appears as 'prompt 1') and change it to 'prompt 0'.
5. Locate the line 'timeout <tenth_of_secs>' (usually appears as 'timeout 600') and change it to 'timeout 0'.

Note: If the timeout line does not exist, **do not** add it.

The *isolinux.cfg* file should now look like the following:

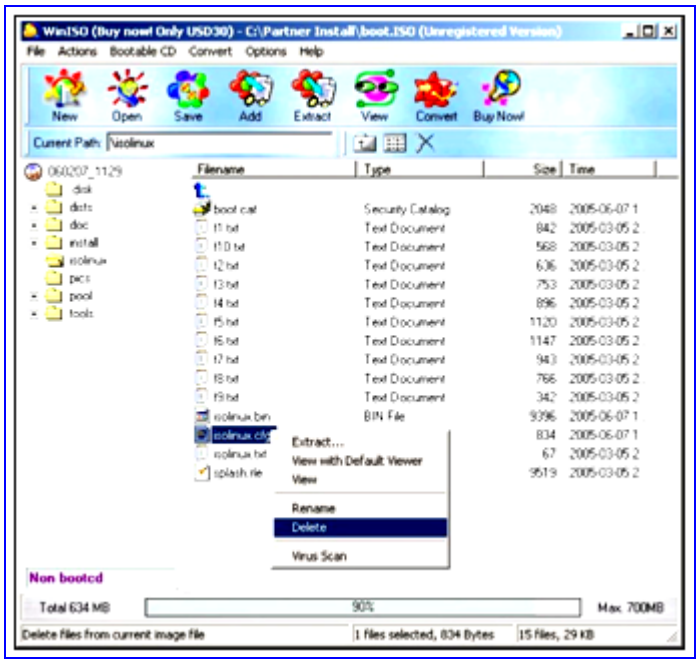
```
serial 0 115200
DEFAULT /install/vmlinuz

APPEND  vga=normal initrd=/install/initrd.gz text
console=ttyS0,115200 ramdisk_size=10240 root=/dev/rd/0
devfs=mount,dall rw --

LABEL linux
kernel /install/vmlinuz
...
DISPLAY isolinux.txt
TIMEOUT 0
PROMPT 0
F1 f1.txt
F2 f2.txt
F3 f3.txt
F4 f4.txt
F5 f5.txt
F6 f6.txt
F7 f7.txt
F8 f8.txt
F9 f9.txt
F0 f10.txt
```

Note: If the timeout line does not exist, do not add it.
1.If you want to install 'kernel 2.6' rather than the 'default 2.4 version', then:
a) Take the options from 'kernel' and 'append' lines under the label called 'LABEL linux26'.
b) Replace the options of 'DEFAULT' and 'APPEND' lines (at the start of the file).
c) Apply the changes from Step 8..

6. Save the changes to the *isolinux.cfg* file, and then close the text editor.
7. Open the 'Partner Install' folder and with the ISO editing utility, open the *boot.iso* file.
8. Click the 'isolinux' directory; the folder's contents appear.

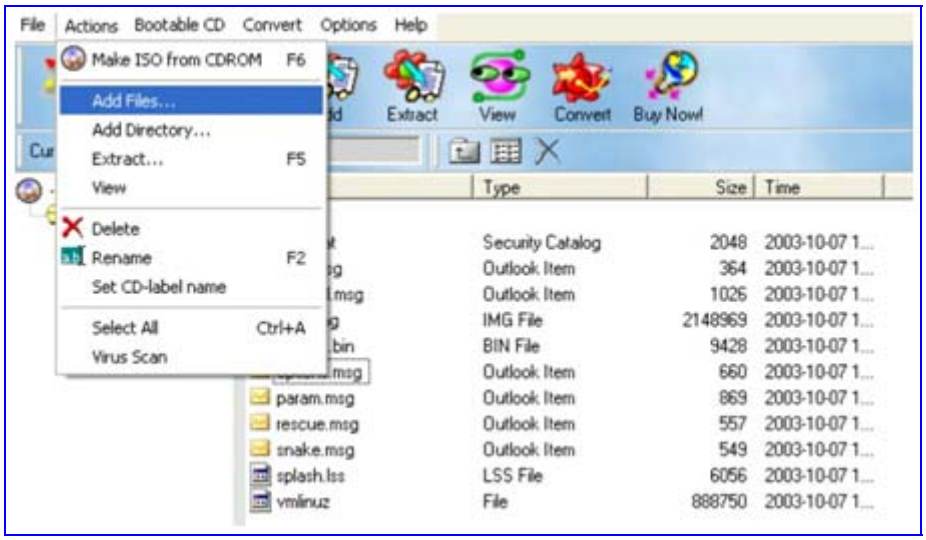
9. Right-click the *isolinux.cfg* file, and then from the shortcut menu, choose **Delete** to delete this file.

Figure 14-19: Deleting CFG



10. From the ISO edit utility menu, select the **Actions** option, followed by **Add Files**.

Figure 14-20: File Add


File Actions Bootable CD Convert Options Help
Make ISO from CDROM F6
Add Files...
Add Directory...
Extract... F5
View
Delete
Rename F2
Set CD-label name
Select All Ctrl+A
Virus Scan
Extract View Convert Buy Now!
Type Size Time
Security Catalog 2048 2003-10-07 1...
Outlook Item 364 2003-10-07 1...
Outlook Item 1026 2003-10-07 1...
IMG File 2148969 2003-10-07 1...
BIN File 9428 2003-10-07 1...
Outlook Item 660 2003-10-07 1...
param.msg Outlook Item 869 2003-10-07 1...
rescue.msg Outlook Item 557 2003-10-07 1...
snake.msg Outlook Item 549 2003-10-07 1...
splash.lss LSS File 6056 2003-10-07 1...
vmlinuz File 888750 2003-10-07 1...

11. Navigate to the 'Partner Install' folder, select the *isolinux.cfg* file, and then click **Open**.

Figure 14-21: ISO Open Function

The updated *isolinux.cfg* file is copied to the 'Partner Install\isolinux' directory.

12. Save the *boot.iso* file in the 'Partner Install' folder.

14.4.4 Stage 4: Burning ISO Image to CD

➢ To burn the boot.iso file to CD, take these 3 steps:

1. Open a burning utility.
2. Use the **Burn Image** option to burn the *boot.iso* file to an empty CD media.
3. Label the media as the “Boot CD”.

Note: Ensure that the *boot.iso* file is burned as an image and not as a data file.

14.4.5 Stage 5: Installing the Boot Media

Now you have the boot media which enables the installation of the Mediant 1000 using the serial connection (terminal) with RS-232 cable.

➢ To complete the installation, take these 7 steps:

1. Connect your Windows™ PC to the Mediant 1000 using a serial cable.
2. Connect the USB CD-ROM device to the Mediant 1000 using a USB cable.
3. Power up the PC.
4. Open the Terminal application (e.g. HyperTerminal) on your Windows™ PC. Create a new connection with the following settings:
   - Connect Port: COM1
   - Baudrate: 115200 (bits per second)
   - Data Bits: 8
   - Parity: None

- Stop Bits: 1
- Flow Control: None

5. Insert the “Boot CD” (created in Stage 3) into the USB CD-ROM drive.
6. Power up the Mediant 1000.
7. On the Terminal application, the BIOS phase starts and the Linux installation begins. The installation uncompresses the kernel, loads it and its drivers, and then starts the interactive installation.

Notes:

- After the BIOS phase, some badly formatted text may appear on the screen.
- From this point on, you should proceed with the screen instructions, as instructed by your Linux distributor.

14.4.6 Additional Linux™ Debian Installation Notes

Please refer to the following notes for the remaining part of the installation.

1. The first interactive screen should be 'Language Selection'. **Before you select the language you wish to use, remove the “Boot CD” and insert the first Debian installation disk**. Select the language you wish to use.
2. As you continue with the installation, you should go through the following stages of the installation:
   a. Country selection
   b. CROM scan
   c. DHCP configuring
3. Configure your network settings manually by selecting the **Configure network manually** option. Either:
   - the DHCP configuration will fail and a screen enabling you to configure the network manually will appear,

   - or -

   - go to the Main Installation menu and select the manual network configuration. Ensure the network is connected and set a **Static IP Address** and all the other network parameters.
4. Continue with the installation, going through the following screens:
   - 'Partition disks'
   - 'Installing the Debian base system'
5. In the 'Install the GRUB boot loader on hard disk' screen, select **Go Back**. This returns you to the 'Debian installer main menu' screen. This is the main menu of the installation.
6. Scroll down in the menu and instead of using GRUB select **Install LILO boot loader on a hard disk**.

7. In the 'LILO Installation target', select **/dev/hda: Master Boot Record**.

Note: The bootloader should detect the serial console. If it doesn't then either:
a) pass additional parameters to the kernel (configure that using the main installation menu)
or/and
b) when configuring LILO you should specify that it tells the kernel to use serial console at speed 115200. In other words append "`text console=ttyS0,115200`" to the kernel options using LILO.
If the serial console wasn't detected during installation, add the following line to the LILO's configuration file after the installation: "`serial=0,115200`"

8. In the next screen, the installation should display the following:

```
LILO is configured to use serial port ttyS0 as the console.
The serial port speed is set to 115200.
```

Select **Continue**; the 'Finish the installation' screen appears followed by 'Installation complete'.

9. Remove the CD and close the CD-ROM drive tray; the system reboots and after the BIOS phase, the kernel starts being uncompressed.

10. The basic installation is now complete. The 'Debian Configuration' screen should appear and this is the phase where you select additional packages to be installed.

Note: Ensure that you install **telnet** and that the **ssh** packages are installed, so that you can connect to your newly-installed system in the future (usually installed by default). From this point on, you should proceed as instructed by your Linux distributor.

11. After the whole installation has been completed, you will be able login to the system from the serial console and/or to "ssh" on your Mediant 1000 (to create an SSH remote connection to it) and to continue its post-installation configuring. You can use the boot media you have created in order to install multiple Mediant 1000 stations.

14.5 Installing Linux™ SUSE

Perform the following five stages for the Linux™ SUSE Installation.

Note: Some distributions of Linux may vary slightly.

14.5.1 Additional Requirement for Linux™ SUSE Installation

To install Linux™ SUSE, a terminal emulation program is required that supports the following:

- ANSI colors (or Linux™ emulation)
- Changing terminal size (to 132x47)

The Tera Term™ program may be used (visit http://hp.vector.co.jp/authors/VA002416/teraterm.html

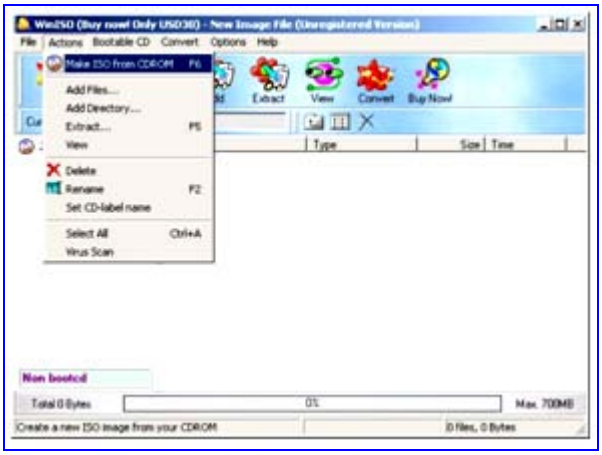
14.5.2 Stage 1: Obtaining the ISO Image

To obtain an updated ISO image, create it using the procedure described below.

➢ To create an ISO image using an ISO editor utility, take these 4 steps:

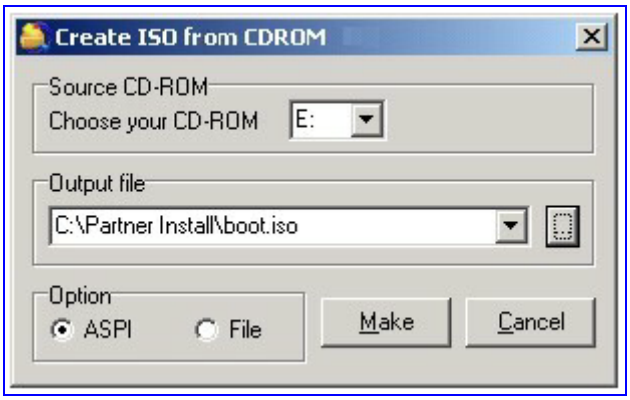
1. Insert the first installation disk of the Linux™ SUSE distribution into the CD-ROM drive of the Windows™ PC.
2. Using Internet Explorer, download a utility that allows editing of an ISO image (e.g., WinISO™ at http://www.winiso.com/download.htm
3. Start WinISO™, and then from the **Actions** menu, select **Make ISO from CDROM**.

Figure 14-22: WinISO - Actions Screen



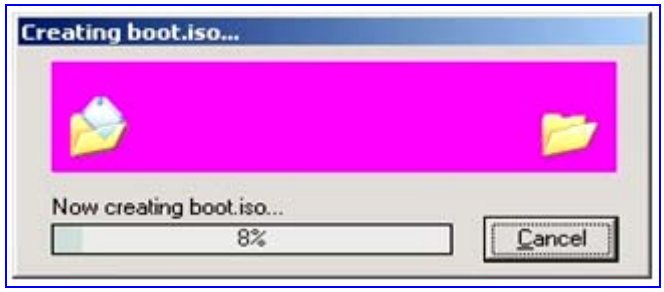
4. Create a 'Partner Install' folder on your hard drive. Select *boot.iso* as the output filename, and then click **Make**.

Figure 14-23: Create ISO from CD-ROM



The utility begins creating the *boot.iso* file.

Figure 14-24: Creating .iso File



14.5.3 Stage 2: Preparing the Boot Media

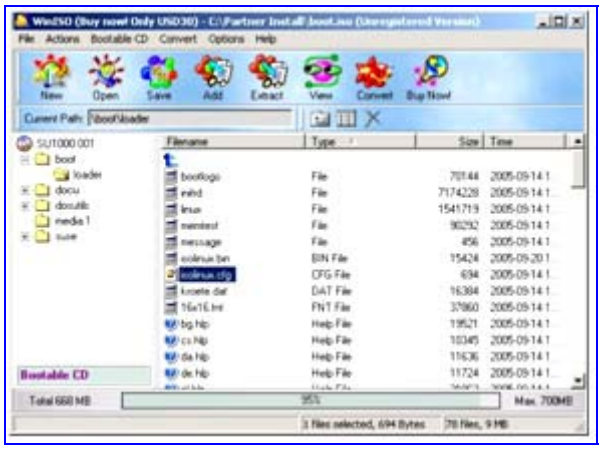
➢ To prepare the Boot Media, take these 5 steps:

1. If you have not already done so, download a utility that allows editing of an ISO image (e.g., WinIso™ at http://www.winiso.com/download.htm

2. Using Internet Explorer, download a UNIX File Format text editor (e.g., PSPad™ at http://www.pspad.comr UltraEdit™ at http://www.ultraedit.com

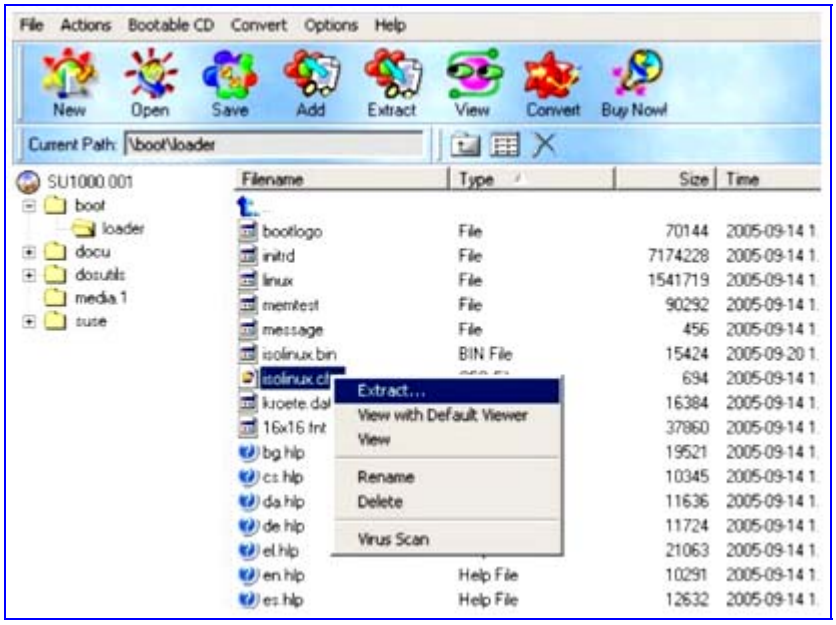
3. Locate the *boot.iso* file in the 'Partner Install' folder on the hard disk of your PC and navigate to the *isolinux.cfg* file.

Figure 14-25: Partner Install Folder



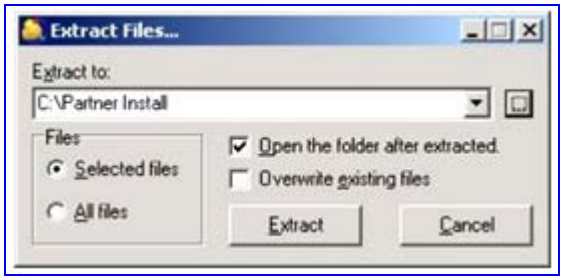
4. Extract the *isolinux.cfg* file by right-clicking the filename, and then from the shortcut menu, choosing **Extract**.

Figure 14-26: Extract isolinux.cfg File



5. Extract the *isolinux.cfg* file to the 'Partner Install' folder.

Figure 14-27: Extracting Files to Partner Install Folder



14.5.4 Stage 3: Editing the isolinux.cfg File

To obtain an updated isolinux.cfg file, perform on of the following:

- Download it from the AudioCodes Web site as described in "Downloading an updated SUSE isolinux.cfg file" on page 547
- Edit it using the steps detailed in "Editing the isolinux.cfg File" on page 548

14.5.4.1 Downloading an Updated SUSE isolinux.cfg File

➢ To download an updated SUSE isolinux.cfg file from AudioCodes Web site, take these 6 steps:

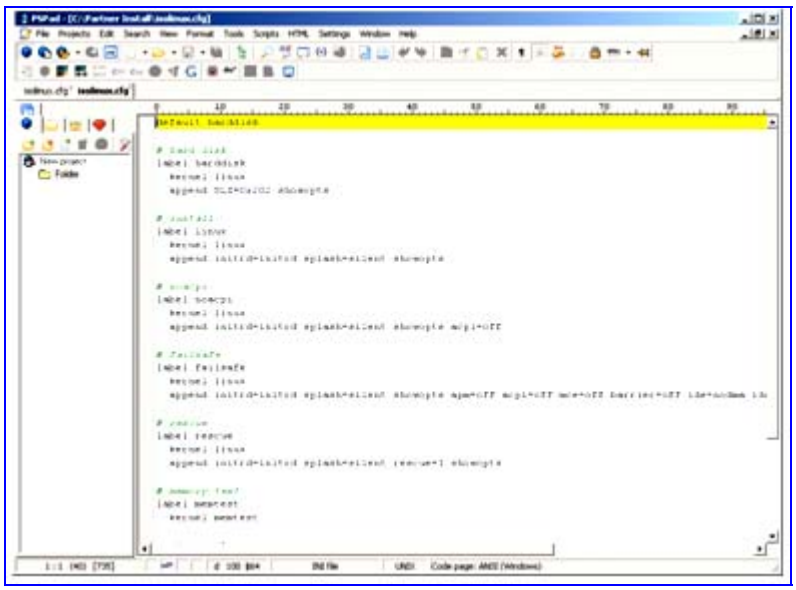
1. Access AudioCodes' Web site (http://www.audiocodes.com), and then navigate to the 'Support' page.
2. Click the **Registered Users Login** link, and then login with your username and password.
3. Under 'Product Documentation', select the **Mediant 1000** link, and then click **Mediant 1000 OSN Server**.
4. Select **Linux Boot Image**, and then select the required installation.
5. Download the *isolinux.cfg* compressed file to a folder called 'Partner Install' on your PC and extract it.
6. Continue with "Editing the isolinux.cfg File" on page 548.

14.5.4.2 Editing the isolinux.cfg File

➢ **To edit the isolinux.cfg file, take these 19 steps:**

1. From the 'Partner Install' folder, open the *isolinux.cfg* file with a text editor that supports UNIX file format (e.g., PSPad or UltraEdit).

Figure 14-28: isolinux.cfg File



2. Insert the following line at the beginning of the file, so that it is the first line.

```
serial 0 115200
```

3. Locate the line 'DEFAULT <my_label>' (usually the first line of the file, e.g., 'default harddisk' or 'default linux'.
4. If <my_label> isn't 'linux' (for instance it can be 'harddisk'), change it to 'linux'.
5. Locate the following line: 'label linux'.
6. Under the 'label linux' line, two lines should appear.

```
kernel...
append...
```

The following examples show how the 'label linux' line and its 'kernel' and 'append' sublines may appear before you change them:

```
label linux
kernel linux
      append initrd=initrd splash=silent showopts
or
label linux
kernel linux
append initrd=initrd ramdisk_size=65536 splash=silent
showopts
```

7. Remove the 'splash=...' and 'showopts' parameters, if they appear in the 'append' line.

Note: Do not remove any other parameters in the 'append' line, especially the 'initrd=' parameter.

8. Add the following parameters to the 'append' line:

```
text console=ttyS0,115200.
```

Note: In the above string, “ttyS0,115200” consists of a capital “S”, only zeros, and one comma.

The following examples show how the 'label linux' line and its 'kernel' and 'append' sublines may appear after you change them:

```
label linux
  kernel linux
  append initrd=initrd text console=ttyS0,115200
or
label linux
  kernel linux
  append initrd=initrd ramdisk_size=65536 text
console=ttyS0,115200
```

9. Locate the line 'prompt <flag>' (usually appears as 'prompt 1') and then change it to 'prompt 0'.

10. Locate the line 'timeout <tenth_of_secs>' (usually appears as 'timeout 200') and then change it to 'timeout 0'.

Note: If the timeout line does not exist, **do not** add it.

The *isolinux.cfg* file should now look like the following:

```
serial 0 115200
default linux

# hard disk
label harddisk
  kernel linux
  append SLX=0x202 showopts

# install
label linux
  kernel linux
  append initrd=initrd text console=ttyS0,115200

# noacpi
label noacpi
  kernel linux
  append initrd=initrd splash=silent showopts acpi=off

# failsafe
label failsafe
  kernel linux
  append initrd=initrd splash=silent showopts apm=off acpi=off
         mce=off barrier=off ide=nodma idewait=50 i8042.nomux
         psmouse.proto=bare irqpoll
# rescue
label rescue
  kernel linux
  append initrd=initrd splash=silent rescue=1 showopts

# memory test
label memtest
  kernel memtest
```

```
implicit     1
gfxboot      bootlogo
display      message
prompt       0
timeout      0
readinfo     2
framebuffer  1
notice       2
```

11. Save the changes to the *isolinux.cfg* file, and then close the text editor.
12. Open the 'Partner Install' folder and with the ISO edit utility, open the *boot.iso* file.
13. Navigate to the *isolinux.cfg* file, right-click it, and then from the shortcut menu, choose **Delete** to delete this file.

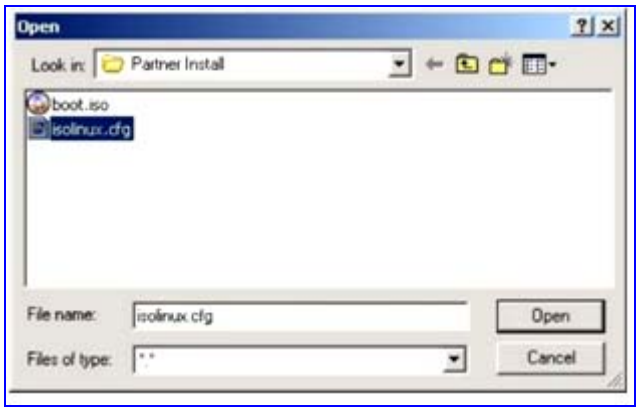
Figure 14-29: Deleting CFG File

14. From the ISO edit **Actions** menu, select **Add Files**.

Figure 14-30: Add CFG File

15. Navigate to the 'Partner Install' folder, select the *isolinux.cfg* file, and then click **Open**.

Figure 14-31: Partner Install Folder



The updated *isolinux.cfg* file is added to the 'Partner Install' folder.

16. Save the *boot.iso* in the 'Partner Install' folder.

Figure 14-32: Save boot.iso

14.5.5 Stage 4: Burning the CD

➢ To burn the CD image, take these 3 steps:

1. Open a burning utility.
2. Use the **Burn Image** option to burn the *boot.iso* to an empty CD media.
3. Mark the media as the “Boot CD”.

Note: Ensure that the *boot.iso* file should be burned as an image and not as a data file.

14.5.6 Stage 5: Installing the Boot Media

Now you have the boot media which enables SUSE installation of the Mediant 1000 using serial connection (terminal) with RS232 cable.

➢ To complete the installation, take these 8 steps:

1. Connect your Windows™ PC to the Mediant 1000 using a serial cable.
2. Open the Terminal application on your Windows™ PC. (Refer to the Additional Requirement in "Additional Requirement for Linux¢ SUSE Installation" on page 544). Create a new connection with the following:
   - Connect Port = COM1
   - Baudrate = 115200 (or bits per second)
   - Data Bits = 8
   - Parity = None
   - Stop Bits = 1
   - Flow Control = None
   - Terminal settings = 132x47, ANSI Color (optional)

Note: If you are using Tera Term, navigate to the "Setup" option in the main menu and use the following configuration.

- Serial port:
  >> Baud rate: 115200
- Terminal Setup:
  >> Terminal size: 132x47
  >> Term size = win size
  >> Terminal ID: VT100
- Window:
  >> Scroll buffer: 47
- Font:
  >> Font: Terminal
  >> Size: 8 (the default size is 10. If the window "slides" beyond your display then set the size to 8).

3. Connect the USB CD-ROM device via USB cable to the Mediant 1000.
4. Insert the "Boot CD" into the USB CD-ROM.
5. Power up the Mediant 1000.

6. On the Terminal screen, the BIOS phase starts and the Linux™ installation begins. The installation uncompresses the kernel, loads it and its drivers and starts the interactive installation.

Notes:

- After the BIOS phase, some badly formatted text may appear on the screen.
- From this point on, you should proceed with the screen instructions as instructed by your Linux distributor.

7. During the installation process, the system may reboot at some stage. In this case, remove the CD from the CD-ROM device.
8. During the installation of the fifth CD, the following items should be configured:
   - Username and password.
   - Firewall configuration: In the 'firewall selection/configuration' screen, select **Firewall: disabled**. In some distributions, you may have to select the **Change** option and set the configuration to 'manual'. This disables the firewall.
   - Network interfaces: The installation sets your network configuration to DHCP by default. If you plan to connect to the system immediately after the installation you'll either have to know the IP address assigned to you by the DHCP or set a **static IP address, subnet mask and default gateway**.

After the whole installation has been completed, you will be able login to the system from the serial console and/or to 'ssh' on your Mediant 1000 (to create an SSH remote connection to it) and to continue its post-installation configuring. You can use the boot media you have created to install multiple Mediant 1000 stations.

Reader's Notes

15 Supplied SIP Software Package

The table below lists the standard SIP software package supplied with the SIP device.

Table 15-1: Supplied Software Package

File Name	Description
Ram.cmp file	
M1000_Digital_SIP_xxx.cmp	Image file containing the software for the Mediant 1000 device.
M1000_SIP_xxx.cmp	Image file containing the software for both FXS and FXO modules.
M600_Digital_SIP_xxx.cmp	Image file containing the software for the Mediant 600 device.
ini files	
SIPgw_M1K.ini	Sample ini file for the Mediant 1000 devices.
M1000_Digital_SIP_T1.ini	Sample ini file for Mediant 1000 E1 devices.
M1000_Digital_SIP_E1.ini	Sample ini file for Mediant 1000 T1 devices.
Coeff_FXO.dat	Telephony interface configuration file for FXO modules.
Coeff_FXS.dat	Telephony interface configuration file for FXS modules.
SIPgw_M1K.ini	Sample ini file for the Mediant 600 devices.
M600_Digital_SIP_T1.ini	Sample ini file for Mediant 600 E1 devices.
M600_Digital_SIP_E1.ini	Sample ini file for Mediant 600 T1 devices.
Usa_tones_xx.dat	Default loadable Call Progress Tones dat file
Usa_tones_xx.ini	Call Progress Tones ini file (used to create dat file)
Utilities	
DConvert	TrunkPack Downloadable Conversion Utility - to create Call Progress Tones, Voice Prompts, and CAS files
ACSyslog	Syslog server
BootP	BootP/TFTP configuration utility
CPTWizard	Call Progress Tones Wizard
CAS Protocol Files	Used for various signaling types, such as E_M_WinkTable.dat
MIB Files	MIB library for SNMP browser
CAS Capture Tool	Utility that is used to convert CAS traces to textual form
ISDN Capture Tool	Utility that is used to convert ISDN traces to textual form

Reader's Notes

16 Selected Technical Specifications

16.1 Mediant 1000

The table below lists the main technical specifications of the Mediant 1000.

Table 16-1: Mediant 1000 Functional Specifications

Function	Specification
Modularity and Capacity	
	Six module slots supporting a mixed configuration of the following interfaces: ▪ Up to 24 FXS/FXO analog ports (up to 6 analog modules) ▪ Up to 4 E1/T1/J1 digital trunks (fully flexible, up to 4 modules) ▪ Up to 20 BRI lines (up to 5 BRI modules) **Note:** Channel capacity depends on configuration settings.
Interface I/O Modules	
FXS Telephony Interface	Up to 6 modules with 4 x FXS RJ-11 ports per module (i.e., up to 24 analog FXS RJ-11 ports)
FXO Telephony Interface	Up to 6 modules with 4 x FXO RJ-11 ports per module (i.e., up to 24 analog FXO RJ-11 ports)
E1/T1/J1 Digital Modules	▪ Up to 4 digital modules (maximum 4 spans per device). Each module supports 1, 2, 3, or 4 E1/T1/J1 spans (Balanced 120/100 Ohm) using RJ-48 connectors ▪ Optional 1+1 or 2+2 fallback
BRI Digital Modules	Up to 5 BRI modules, each supporting up to 4 BRI ports (maximum 20 BRI ports) using RJ-45 connectors
FXS Functionality	
FXS Capabilities	▪ Short or long haul, up to 3,000 m (10,000 ft.), using a 24-AWG line cord ▪ 2 or 4 ports per FXS module ▪ Caller ID generation: Bellcore GR-30-CORE Type 1 using Bell 202 FSK modulation, ETSI Type 1, NTT, Denmark, India, Brazil, British and DTMF ETSI CID (ETS 300-659-1) ▪ Polarity Reversal and Wink signals generation ▪ Message waiting indication (lamp) ▪ Programmable Line Characteristics: Battery feed, line current, hook thresholds, AC impedance matching, hybrid balance, Tx & Rx frequency response, Tx & Rx Gains **Note:** For a specific coefficient file, please contact AudioCodes. ▪ Configurable ringing signal: up to four cadences and frequency 10 to 200 Hz. ▪ Drive 5 phones per port simultaneously in offhook and Ring states. REN = 5. ▪ Over-temperature protection for abnormal situations such as shorted lines ▪ Lifeline on every FXS factory-preconfigured module

Function	Specification
FXO Functionality	
FXO Capabilities	▪ Short or long haul, up to 7,000 m (24,000 ft.), using 24 AWG line cord ▪ 4 ports per FXO module ▪ Far-end disconnect detection ▪ Lightning and high voltage protection for outdoor operation ▪ Programmable Line Characteristics: AC impedance matching, hybrid balance, Tx & Rx frequency response, Tx & Rx Gains, ring detection threshold, DC characteristics **Note:** For a specific coefficient file, please contact AudioCodes. ▪ Caller ID Detection: Bellcore GR-30-CORE Type 1 using Bell 202 FSK modulation, ETSI Type 1, NTT, Denmark, India, Brazil, and DTMF ETSI CID (ETS 300-659-1) ▪ Polarity Reversal and Wink signal detection
Voice & Tone Characteristics	
Voice Compression	G.711 PCM at 64 kbps µ-law/A-law; EG.711 µ-law/A-law at 64 kbps; G.723.1 MP-MLQ at 5.3 or 6.3 kbps; G.726 at 32 kbps ADPCM; G.729 CS-ACELP 8 Kbps Annex A / B; NetCoder at 6.4, 7.2, 8.0 and 8.8 kbps; iLBC; Microsoft GSM (40 msec)
Silence Suppression	▪ G.723.1 Annex A ▪ G.729 Annex B ▪ PCM and ADPCM - Standard Silence Descriptor (SID) with Proprietary Voice Activity Detection (VAD) and Comfort Noise Generation (CNG)
Packet Loss Concealment	G.711 appendix 1; G.723.1; G.729 a/b
Echo Canceler	G.165 and G.168 2000, 64 msec
Gain Control	Configurable
DTMF Transport (in-band)	Mute, transfer in RTP payload or relay in compliance with RFC 2833
DTMF Detection and Generation	Dynamic range 0 to -25 dBm, compliant with TIA 464B and Bellcore TR-NWT-000506
Call Progress Tone Detection and Generation	32 tones: single tone, dual tones or AM tones, programmable frequency & amplitude; 64 frequencies in the range 300 to 1980 Hz, 1 to 4 cadences per tone, up to 4 sets of ON/OFF periods
Output Gain Control	-32 dB to +31 dB in steps of 1 dB
Input Gain Control	-32 dB to +31 dB in steps of 1 dB
Conferencing	
Module	Optional Media Processing Module (MPM) housed in Slot 6 on the chassis' front panel
Conference Channels (Max.)	60
Simultaneous 3-Way Conferences (Max.)	20

Function	Specification
Full-Duplex Parties per Conference Bridge (Max.)	60
Fax/Modem Relay	
Fax Relay	▪ Group 3 fax relay up to 14.4 kbps with auto fallback ▪ T.30 (PSTN) and T.38 (IP) compliant, real time fax relay ▪ Tolerant network delay (up to 9 seconds round trip) ▪ CNG tone detection & Relay per T.38 ▪ Answer tone (CED or AnsAm) detection & Relay per T.38
Fax Transparency	Automatic fax bypass (pass-through) to G.711, ADPCM or NSE bypass mode
Modem Transparency	▪ Automatic switching to PCM or ADPCM on V.34 or V.90 modem detection ▪ Automatic switching (pass-through) to PCM, ADPCM or NSE bypass mode for modem signals (V.34 or V.90 modem detection)
Protocols	
VoIP Signaling Protocol	SIP RFC 3261
Communication Protocols	▪ RTP/RTCP packetization ▪ IP stack (UDP, TCP, and RTP) ▪ Remote Software load (TFTP, HTTP and HTTPS)
Telephony Protocols	▪ PRI (ETSI Euro ISDN, ANSI NI2, 4/5ESS, DMS 100, QSIG, Japan INS1500, Australian Telecom, New Zealand Telecom, Hong Kong Variant, Korean MIC) E1/T1 CAS protocols: MFC R2, E&M wink start ▪ BRI protocols (ETSI 5ESS and QSIG over BRI) ▪ Immediate start, delay start, loop start, ground start ▪ Feature Group B, D for E1/T1
In-Band Signaling	▪ DTMF (TIA 464A) ▪ MF-R1, MFC R2 ▪ User-defined Call Progress Tones
Line Signaling Protocols	Loop start and ground start The standard FXO module supports outdoor and indoor (lightning protection) loop-start signaling. The **FXO G** module supports both loop- and ground-start signaling (but only supports indoor protection).
CPU	
Network Interface	Two Ethernet RJ-45 connectors, 10/100Base-TX
RS-232 Interface	RS-232 terminal interface. Non-standard RS-232 connector on the device's CPU.
Reset	Resets the device
Dry Contact	NB (Night Bell) and paging
Audio I/O	MOH (Music on Hold) and paging

Function	Specification
Physical	
Dimensions (W x H x D)	482.6 mm (19") x 1U x 350.5 mm (13.8")
Weight	Approx. 5 kg (depending on number of installed modules)
Supply Voltage and Power Consumption	Universal 100 - 240 VAC; 50 - 60 Hz; 1 A max.
Environmental	▪ Operational: 0 to 45°C (32 to 113°F) ▪ Storage: -10 to 70°C (14 to 158°F) ▪ Humidity: 10 to 90% non-condensing
Installation	Standard 19-inch rack mount or shelf
OSN Server	
Single Chassis Integration	Embedded Pentium™ Celeron™ based platform for third-party hosted applications
CPU	▪ Intel TM Pentium-M Celeron ULV 600 MHz processor ▪ 855 GME + 6300 ESB Intel chipset ▪ 128 KB BIOS flash ▪ Up to 1 Gigabyte 200/266 MHz SODIMM memory ▪ Operational temperature: 0 to 40°C ▪ Dimensions (W x H x D): 75 mm (2.9") x 30 mm (1.17") x 160 mm (6.4")
HHD	40 Gigabyte Hard Disk Drive 5200 RPM (second HDD optional)
Interfaces	10/100Base-TX, USB, RS-232, NB relay, MOH
Diagnostics	
Front panel Status LEDs	▪ E1/T1 status ▪ LAN status ▪ Status of device (Fail, ACT, Power, and Swap Ready)
Syslog events	Syslog Server (RFC 3164 IETF standard)
SNMP MIBs and Traps	SNMP v2c, SNMP v3
Management	
Configuration	Embedded Web Server (Web browser), CLI or *ini* file
Management and Maintenance	▪ SNMP v2c, SNMP v3 ▪ Syslog (according to RFC 3164) ▪ Local RS-232 terminal ▪ Web Management via HTTP or HTTPS ▪ Telnet

16.2 Mediant 600

The table below lists the main technical specifications of the Mediant 600.

Table 16-2: Mediant 600 Functional Specifications

Function	Specification
Interfaces and Capacity	
E1/T1/J1 Digital Modules	▪ 1 or 2 E1/T1/J1 spans using RJ-48c ports ▪ 1 Fractional E1/T1/J1 (15 DS0) span using RJ-48c port (Balanced 120/100 Ohm) ▪ Optional 1+1 PSTN/Lifeline fallback
BRI Digital Modules	4 or 8 BRI ports (RJ-45 ports) with S/T interfaces (4 BRI ports per module), providing up to 16 calls
Voice & Tone Characteristics	
Voice Compression	G.711 PCM at 64 kbps µ-law/A-law; EG.711 µ-law/A-law at 64 kbps; G.723.1 MP-MLQ at 5.3 or 6.3 kbps; G.726 at 32 kbps ADPCM; G.729 CS-ACELP 8 Kbps Annex A / B; NetCoder at 6.4, 7.2, 8.0 and 8.8 kbps; iLBC; Microsoft GSM (40 msec)
Silence Suppression	▪ G.723.1 Annex A ▪ G.729 Annex B ▪ PCM and ADPCM - Standard Silence Descriptor (SID) with Proprietary Voice Activity Detection (VAD) and Comfort Noise Generation (CNG)
Packet Loss Concealment	G.711 appendix 1; G.723.1; G.729 a/b
Echo Canceler	G.165 and G.168 2000, 64 msec
Gain Control	Configurable
DTMF Transport (in-band)	Mute, transfer in RTP payload or relay in compliance with RFC 2833
DTMF Detection and Generation	Dynamic range 0 to -25 dBm, compliant with TIA 464B and Bellcore TR-NWT-000506
Call Progress Tone Detection and Generation	32 tones: single tone, dual tones or AM tones, programmable frequency & amplitude; 64 frequencies in the range 300 to 1980 Hz, 1 to 4 cadences per tone, up to 4 sets of ON/OFF periods
Output Gain Control	-32 dB to +31 dB in steps of 1 dB
Input Gain Control	-32 dB to +31 dB in steps of 1 dB
Fax/Modem Relay	
Fax Relay	▪ Group 3 fax relay up to 14.4 kbps with automatic fallback ▪ T.30 (PSTN) and T.38 (IP) compliant, real time fax relay ▪ Tolerant network delay (up to 9 seconds round trip) ▪ CNG tone detection & Relay per T.38 ▪ Answer tone (CED or AnsAm) detection & Relay per T.38
Fax Transparency	Automatic fax bypass (pass-through) to G.711, ADPCM or NSE bypass mode

Function	Specification
Modem Transparency	▪ Automatic switching to PCM or ADPCM on V.34 or V.90 modem detection ▪ Automatic switching (pass-through) to PCM, ADPCM or NSE bypass mode for modem signals (V.34 or V.90 modem detection)
Protocols	
VoIP Signaling Protocol	SIP RFC 3261
Communication Protocols	▪ RTP/RTCP packetization ▪ RTCP-XR ▪ IP stack (UDP, TCP, and RTP) ▪ Remote software upload (TFTP, HTTP and HTTPS)
Telephony Protocols	▪ PRI (ETSI Euro ISDN, ANSI NI2, 4/5ESS, DMS 100, QSIG, Japan INS1500, Australian Telecom, New Zealand Telecom, Hong Kong Variant, Korean MIC) ▪ VN,3VN4,VN6 ▪ E1/T1 CAS protocols: MFC R2, E&M wink start ▪ BRI protocols: ETSI 5ESS, QSIG over BRI ▪ Immediate start, delay start, loop start, ground start ▪ Feature Group B, D for E1/T1
In-Band Signaling	▪ DTMF (TIA 464A) ▪ MF-R1, MFC R2 ▪ User-defined Call Progress Tones
Line Signaling Protocols	Loop-start and ground-start signaling
CPU	
Network Interface	Two Ethernet 10/100Base-TX ports (RJ-45)
RS-232 Interface	RS-232 terminal interface (non-standard RS-232 connector)
Reset	Reset button for hardware reset
Dry Contact	NB (Night Bell) and paging
Audio I/O	MOH (Music on Hold) and paging
Physical	
Dimensions (W x H x D)	482.6 mm (19”) x 1U x 350.5 mm (13.8”)
Weight	Approx. 5 kg (depending on number of installed modules)
Supply Voltage and Power Consumption	Universal 100 - 240 VAC; 50 - 60 Hz; 1A max.
Environmental	▪ Operational: 0 to 45°C (32 to 113°F) ▪ Storage: -10 to 70°C (14 to 158°F) ▪ Humidity: 10 to 90% non-condensing
Installation	Standard 19-inch rack mount or shelf mount

Function	Specification
Diagnostics	
Front Panel Status LEDs	E1/T1, BRI, and LAN status
Syslog Events	Supported by Syslog Server (RFC 3164 IETF standard)
SNMP MIBs and Traps	SNMP v2c, SNMP v3
Management	
Configuration	Embedded HTTP Web Server (Web browser), CLI or *ini* file
Management and Maintenance	▪ SNMP v2c, SNMP v3 ▪ Syslog (according to RFC 3164) ▪ Local RS-232 terminal ▪ Web Management via HTTP or HTTPS ▪ Telnet
Type Approvals	
Telecommunication Standards	TIA/EIA-IS-968, TBR-4, TBR-13, and TBR-21
Safety and EMC Standards	▪ UL60950-1; FCC 47 CFR part 15 Class B ▪ CE Mark: EN55022 Class B, EN60950-1, EN55024, EN300 386, EN61000-3-2/3-3
Environmental	ETS 300019-2-1 Storage T1.2, ETS 300019-2-2 Transportation T2.3 ETS 300019-2-3 Operating T3.2

Reader's Notes

17 Glossary

Table 17-1: Glossary of Terms

Term	Meaning
ADPCM	Adaptive Differential PCM - voice compression
AIS	Alarm Indication Signal
A-law	Standard companding algorithm, used in European digital communications systems to optimize the dynamic range of an analog signal for digitizing.
AOR	Address of Record
AWG	American Wire Gauge
bps	Bits per second
BootP	AudioCodes Proprietary Bootstrap Loader Utility
BRI	Basic Rate Interface (ISDN)
CAS	Channel Associated Signaling
CoS	Class of Service
CMP	Compressed File (device Firmware)
CPT	Call Progress Tones
dB	Decibels
DHCP	Dynamic Host Control Protocol
DID	Direct Inward Dial
DiffServ	Differentiated Services
DNS	Domain Name System (or Server)
DR	Debug Recording
DS1	1.544 Mbps USA Digital Transmission System (see E1 and T1)
DSP	Digital Signal Processor (or Processing)
DTMF	Dual Tone Multiple Frequency (Touch Tone)
E1	2.048 Mbps European Digital Transmission System (see T1)
ETSI	European Telecommunications Standards Institute
FQDN	Fully Qualified Domain Name
FXS	Foreign Exchange Station
FXO	Foreign Exchange Office
GRUU	Globally Routable User Agent URIs
ICMP	Internet Control Message Protocol
IE	Information Element (ISDN layer 3 protocol, basic building block)
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange (for IPSec)
IP	Internet Protocol

Term	Meaning
IPSec	IP Security
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
ITU	International Telecommunications Union
ITU-T	Telecommunications section of the ITU
IVR	Interactive Voice Response
Jitter	Variation of interpacket timing interval
kbps	Kilobit per second. 1,000 bits per second
LAPD	Line Access Protocol for the D-channel
LFA	Loss of Frame Alignment
LOF	Loss of Frame
Mbps	Megabit per second. Million bits per second
MIB	Management Information Base
MLPP	Multilevel Precedence and Preemption
ms or msec	Millisecond; a thousandth part of a second
MSCML	Media Server Control Markup Language
NT	Network Termination (ISDN)
MWI	Message Waiting Indicator
NAPTR	Naming Authority Pointer
NAT	Network Address Translation
NetCoder	AudioCodes Proprietary High Quality, Speech Coder
NFAS	Non-Facility Associated Signalling (ISDN PRI)
NFS	Network File System
NPI	Numbering Plan Indicator
NTP	Network Time Protocol
OAMP	Operations, Administration, Maintenance and Provisioning
OSI	Open Systems Interconnection (Industry Standard)
PBX	Private Branch Exchange
PCI	Personal Computer Interface (Industry Standard)
PCM	Pulse-Code Modulation
PI	Progress Indicator
PKI	Public-Key Infrastructures
POTS	Plain Old Telephone System or Service
PRT	Prerecorded Tones (File)
PRI	Primary Rate Interface (ISDN)
PSTN	Public Switched Telephone Network

Term	Meaning
PVID	Port VLAN ID (VLAN ID assignment to Ethernet packet by switch)
QoS	Quality of Service
RFC	Request for Comment issued by IETF
RTCP	Real-Time Transport (RTP) Control Protocol
RTP	Real-Time Transport Protocol
SA	Security Associations (contains encryption keys and profile used by IPSec to encrypt the IP stream)
SAS	Stand Alone Survivability Feature
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SMDI	Simplified Message Desk Interface
SME	Small and Medium-sized Enterprise
SNMP	Simple Network Management Protocol
SRTP	Secure Real-Time Transport Protocol
SRV	Service Record
SSH	Secure Shell
SSL	Secure Socket Layer (also known as Transport Layer Security (TLS))
STUN	Simple Traversal of UDP through NATs
T1	1.544 Mbps USA Digital Transmission System (see E1 and DS1)
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol
TE	Terminal Equipment (ISDN)
TDM	Time-Division Multiplexing
TFTP	Trivial File Transfer Protocol
TLS	Transport Layer Security
TON	Type of Numbering
UA	SIP User Agent
UDP	User Datagram Protocol
URI (SIP URIs)	SIP Uniform Resource Indicators
VBD	Voice-band data
VLAN	Virtual Local Area Network
VoIP	Voice over Internet Protocol
VoP	Voice over Packet(s)
VP	Voice Prompts (File)
VPN	Virtual Private Network
µ-Law	A companding algorithm, used in the digital telecommunication systems

AudioCodes Mediant™ Media Gateways

SIP

Mediant 1000 & Mediant 600

User's Manual

Version 5.4

AudioCodes

www.audiocodes.com